02 0197
YOR

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

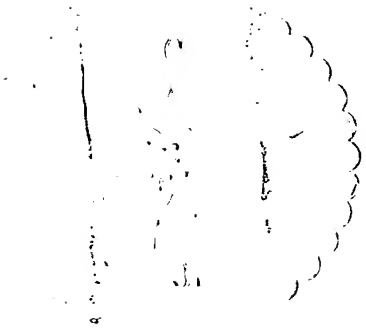
This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００２年１２月　６日

出願番号
Application Number: 特願２００２－３５５６４１

[ST.10/C]: [ＪＰ２００２－３５５６４１]

出願人
Applicant(s): インターナショナル・ビジネス・マシーンズ・コーポレーション



２００３年　５月２０日

特許庁長官
Commissioner,
Japan Patent Office 太田信一郎

出証番号　出証特２００３－３０３７００７

【書類名】　特許願

【整理番号】　JP9020197

【あて先】　特許庁長官殿

【国際特許分類】　G06F　9/44
G06F 13/00

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　山本　学

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和ソフトウェア開発研究所内

【氏名】　小林　敬明

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和ソフトウェア開発研究所内

【氏名】　青木　淳一

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和ソフトウェア開発研究所内

【氏名】　宮島　広之

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　田井　秀樹

【特許出願人】

【識別番号】　390009531

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】　100086243

【弁理士】

【氏名又は名称】　坂口　博

【代理人】

【識別番号】　100091568

【弁理士】

【氏名又は名称】　市位　嘉宏

【代理人】

【識別番号】　100108501

【弁理士】

【氏名又は名称】　上野　剛史

【復代理人】

【識別番号】　100085408

【弁理士】

【氏名又は名称】　山崎　隆

【手数料の表示】

【予納台帳番号】　117560

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9706050

【包括委任状番号】　9704733

【包括委任状番号】　0207860

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　メッセージ処理装置、メッセージ処理方法、及びメッセージ処理プログラム

【特許請求の範囲】

【請求項１】　エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能でとなる複数個のエージェント、

前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込み手段、

メッセージが挿入されているメッセージ・キューに係るエージェントにその作動を指示するエージェント用指示手段、及び

前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込み手段にその処理の繰返しを指示する繰返し指示手段、

を有していることを特徴とするメッセージ処理装置。

【請求項２】　各エージェントは、該エージェントに対応付けられているメッセージ・キューにおけるメッセージに対して、該エージェントに対応付けられ

た処理要求元への通知を実施するものであることを特徴とする請求項１記載のメッセージ処理装置。

【請求項３】　エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能でとなる複数個のエージェント、

第１のメッセージ・キュー用処理機構、

第２のメッセージ・キュー用処理機構、

第１及び第２のメッセージ・キュー用処理機構のどちらかを選択する選択手段、及び

メッセージが挿入されているメッセージ・キューに係るエージェントについて該エージェントがキャッシュ・メモリに有れば該エージェントにその作動を直ちに指示しまた該エージェントがキャッシュ・メモリに無ければ該エージェントを前記永続記憶装置から前記キャッシュ・メモリへ読込んでから該エージェントにその作動を指示するエージェント用指示手段、

を有し、

第１のメッセージ・キュー用処理機構は、

前記リスト情報に含まれる全部の処理要求元に係るメッセージ・キューに前記メッセージを挿入する挿入手段、

を有し、

第２のメッセージ・キュー用処理機構は、

前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込み手段、及び

前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込み手段にその処理の繰返しを指示する繰返し指示手段、

を有している、

ことを特徴とするメッセージ処理装置。

【請求項４】　前記選択手段の選択はオペレータの指示に基づくことを特徴とする請求項３記載のメッセージ処理装置。

【請求項５】　前記選択手段の選択は、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元の予測数、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元に係るエージェントについてのキャッシュ・メモリにおける予測ヒット率、前記選択手段が第１のメッセージ・キュー用処理機構を選択している場合に前記受付手段が情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェントのメッセージ・キューに挿入するまでの予測作業時間、前記選択手段が前記第２のメッセージ・キュー用処理機構を選択している場合に前記受付手段が情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェントのメッセージ・キューに挿入するまでの予測作業時間、キャッシュ・メモリ内のエージェントについてその使用を決定してから該決定したエージェントが処理完了するまでの予測時間、及び／又はキャッシュ・メモリ外のエージェントについてその使用を決定してから該決定したエージェントが処理完了するまでの予測時間に基づくことを特徴とする請求項３記載のメッセージ処理装置。

【請求項６】　エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元を前記処理要求元検索情報に基づいて決定する処理要求元決定手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているとき作動可能となる複数個のエージェント、

各メッセージについての処理優先度を単一の価値基準に基づいてサブ処理優先度として決定する少なくとも１個のサブ処理優先度決定手段、

前記サブ処理優先度決定手段の総個数が２以上であるときは各サブ処理優先度決定手段が各メッセージについて個々に決定したサブ処理優先度に基づいて各メッセージについての複合処理優先度を決定し、また、前記サブ処理優先度決定手段の総個数が１であるときは該唯一のサブ処理優先度決定手段が各メッセージについて決定したサブ処理優先度を複合処理優先度として決定する複合処理優先度決定手段、

各メッセージ・キューが保持しているメッセージの中で最高複合処理優先度のメッセージを最優先メッセージとし該最優先メッセージの複合処理優先度が同一であるメッセージ・キューに係るエージェント同士では、キャッシュ・メモリに存在する方のエージェントを、存在しない方のエージェントより優先してその作動を指示するエージェント用指示手段、

を有していることを特徴とするメッセージ処理装置。

【請求項７】　前記価値基準には、メッセージの内容に係るもの及びメッセージが適用される処理要求元に係るものがあること特徴とする請求項６記載のメッセージ処理装置。

【請求項８】　メッセージの内容に係る所定の価値基準には、メッセージの処理の緊急性に係る価値基準が含まれることを特徴とする請求項７記載のメッセージ処理装置。

【請求項９】　メッセージが適用される処理要求元に係る価値基準には、処理要求元の格付けに係る価値基準が含まれることを特徴とする請求項７記載のメ

ッセージ処理装置。

【請求項１０】　前記エージェント用指示手段により処理を一旦、開始したエージェントは、該エージェントに係るメッセージ・キューが保持するメッセージが複数個あるとき、それら全部のメッセージについて、又は複合処理優先度が上位から所定番目までのメッセージについて連続処理することを特徴とする請求項６記載のメッセージ処理装置。

【請求項１１】　前記サブ処理優先度決定手段及び前記複合処理優先度決定手段を含むエージェント管理手段を有し、

前記エージェント管理手段は、各エージェントが永続記憶装置に存在するか否かを検出する存在検出手段、該存在検出手段の判定結果及び各エージェントの複合処理優先度に基づいてエージェントをグループ化しグループ化情報を管理するグループ化情報管理手段、及びグループ化情報管理手段にグループ化情報の更新を指示する更新指示手段を有し、

前記エージェント用指示手段は、前記エージェント管理手段におけるグループ化情報に基づく順番でエージェントにその作動を指示する、

ことを特徴とする請求項６記載のメッセージ処理装置。

【請求項１２】　エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記受付手段が受付けたエージェント起動原因イベントについての受付順番情報を管理する受付順番情報管理手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能となっている複数個のエージェント、

相互に並列動作可能である複数個のスレッドであって各スレッドはエージェント起動原因イベントを起因とするメッセージが適用される処理要求元を処理要求

元検索情報に基づいて検出しかつ各検出処理要求元に係るメッセージ・キューに前記メッセージを挿入する複数個のスレッド、

各スレッドにそれが処理を担当するエージェント起動原因イベントを割当てる割当て手段、

前記受付手段が受付けた各エージェント起動原因イベントについてのスレッドによる処理の進行状態情報を管理する進行状態情報管理手段、

処理進行状態情報がスレッド処理終了に係る情報になっているエージェント起動原因イベント（以下、「被判定エージェント起動原因イベント」と言う。）について、該被判定エージェント起動原因イベントより前に前記受付手段において受付けたエージェント起動原因イベントの中にまだスレッド処理の未終了になっているエージェント起動原因イベントが有るか無いかを判定する判定手段、及び

前記判定手段が「有る」と判定した被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェントによる処理を抑制するエージェント制御手段、

を有していることを特徴とするメッセージ処理装置。

【請求項１３】　前記判定手段が「無い」と判定した被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェントによる処理を許容する前記エージェント制御手段、

を有していることを特徴とする請求項１２記載のメッセージ処理装置。

【請求項１４】　「無い」と判定したエージェント起動原因イベントに対して受付順番がその次のエージェント起動原因イベントが、「有る」との判定済みのものであるときは、判定結果を「有る」から「無い」へ変更する前記判定手段

を有していることを特徴とする請求項１３記載のメッセージ処理装置。

【請求項１５】　前記判定手段による判定結果が「無い」となっている複数個のエージェント起動原因イベントを生成起因とするメッセージが受付順番方向へ連続するメッセージ･キューについての処理を実行する場合は、それら連続する複数個のメッセージについての処理を連続的に行う前記エージェント、

を有していることを特徴とする請求項１４記載のメッセージ処理装置。

【請求項１６】　エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能であり各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能であるエージェント設定ステップ、

前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込みステップ、

メッセージが挿入されているメッセージ・キューに係るエージェントにその作動を指示するエージェント用指示ステップ、及び

前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込みステップの繰返しを指示する繰返し指示ステップ、

を有していることを特徴とするメッセージ処理方法。

【請求項１７】　各エージェントは、該エージェントに対応付けられているメッセージ・キューにおけるメッセージに対して、該エージェントに対応付けられた処理要求元への通知を実施するものであることを特徴とする請求項１６記載のメッセージ処理装置。

【請求項１８】　エージェント起動原因イベントに対して該当処理要求元を

検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能であり各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能であるエージェント設定ステップ、

第１のメッセージ・キュー用処理ステップ、

第２のメッセージ・キュー用処理ステップ、

第１及び第２のメッセージ・キュー用処理ステップのどちらかを選択する選択ステップ、及び

メッセージが挿入されているメッセージ・キューに係るエージェントについて該エージェントがキャッシュ・メモリに有れば該エージェントにその作動を直ちに指示しまた該エージェントがキャッシュ・メモリに無ければ該エージェントを前記永続記憶装置から前記キャッシュ・メモリへ読込んでから該エージェントにその作動を指示するエージェント用指示ステップ、

を有し、

第１のメッセージ・キュー用処理ステップは、

前記リスト情報に含まれる全部の処理要求元に係るメッセージ・キューに前記メッセージを挿入する挿入ステップ、

を有し、

第２のメッセージ・キュー用処理ステップは、

前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・

キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込みステップ、及び

前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込みステップの繰返しを指示する繰返し指示ステップ、

を有している、

ことを特徴とするメッセージ処理方法。

【請求項１９】　前記選択ステップにおける選択はオペレータの指示に基づくことを特徴とする請求項１８記載のメッセージ処理方法。

【請求項２０】　前記選択ステップにおける選択は、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元の予測数、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元に係るエージェントについてのキャッシュ・メモリにおける予測ヒット率、第１のメッセージ・キュー用処理ステップにおいて処理を割当てられた場合に前記受付ステップにおいて情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェントのメッセージ・キューに挿入するまでの予測作業時間、第２のメッセージ・キュー用処理ステップにおいて処理を割当てられた場合に前記受付ステップにおいて情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェントのメッセージ・キューに挿入するまでの予測作業時間、キャッシュ・メモリ内のエージェントについてその使用を決定してから該決定したエージェントが処理完了するまでの予測時間、及び／又はキャッシュ・メモリ外のエージェントについてその使用を決定してから該決定したエージェントが処理完了するまでの予測時間に基づくことを特徴とする請求項１８記載のメッセージ処理方法。

【請求項２１】　エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ

エージェント起動原因イベントを受付ける受付ステップ、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元を前記処理要求元検索情報に基づいて決定する処理要求元決定ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは各処理要求元に対応付けられており永続記憶装置に記憶され各エージェントは永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能でありかつ各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能となっているように設定するエージェント設定ステップ、

各メッセージについての処理優先度を単一の価値基準に基づいてサブ処理優先度として決定する少なくとも１個のサブ処理優先度決定ステップ、

前記サブ処理優先度決定ステップの総個数が２以上であるときは各サブ処理優先度決定ステップにおいて各メッセージについて個々に決定したサブ処理優先度に基づいて各メッセージについての複合処理優先度を決定し、また、前記サブ処理優先度決定ステップの総個数が１であるときは該唯一のサブ処理優先度決定ステップにおいて各メッセージについて決定したサブ処理優先度を複合処理優先度として決定する複合処理優先度決定ステップ、及び

各メッセージ・キューが保持しているメッセージの中で最高複合処理優先度のメッセージを最優先メッセージとし該最優先メッセージの複合処理優先度が同一であるメッセージ・キューに係るエージェント同士では、キャッシュ・メモリに存在する方のエージェントを、存在しない方のエージェントより優先してその作動を指示するエージェント用指示ステップ、

を有していることを特徴とするメッセージ処理方法。

【請求項２２】　前記価値基準には、メッセージの内容に係るもの及びメッセージが適用される処理要求元に係るものがあること特徴とする請求項２１記載のメッセージ処理方法。

【請求項２３】　メッセージの内容に係る所定の価値基準には、メッセージ

の処理の緊急性に係る価値基準が含まれることを特徴とする請求項２２記載のメッセージ処理方法。

【請求項２４】　メッセージが適用される処理要求元に係る価値基準には、処理要求元の格付けに係る価値基準が含まれることを特徴とする請求項２２記載のメッセージ処理方法。

【請求項２５】　前記エージェント用指示ステップにより処理を一旦、開始したエージェントは、該エージェントに係るメッセージ・キューが保持するメッセージが複数個あるとき、それら全部のメッセージについて、又は複合処理優先度が上位から所定番目までのメッセージについて連続処理することを特徴とする請求項２１記載のメッセージ処理方法。

【請求項２６】　前記サブ処理優先度決定ステップ及び前記複合処理優先度決定ステップを含むエージェント管理ステップを有し、

前記エージェント管理ステップは、各エージェントが永続記憶装置に存在するか否かを検出する存在検出ステップ、及び該存在検出ステップの判定結果及び各エージェントの複合処理優先度に基づいてエージェントをグループ化しグループ化情報を管理するとともに該グループ化情報を適宜更新するグループ化情報管理ステップを有し、

前記エージェント用指示ステップは、前記エージェント管理ステップにおけるグループ化情報に基づく順番でエージェントにその作動を指示する、

ことを特徴とする請求項２１記載のメッセージ処理方法。

【請求項２７】　エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記受付ステップにおいて受付けたエージェント起動原因イベントについての受付順番情報を管理する受付順番情報管理ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは、各処理要求元に対応付けられており、永続記憶装置に記憶され、永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及び

キャッシュ・メモリから破棄可能であり、かつ各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能となっているように設定するエージェント設定ステップ、

複数個のスレッドを設定するスレッド設定ステップであって各スレッドは、相互に並列動作可能であり、エージェント起動原因イベントを起因とするメッセージが適用される処理要求元を処理要求元検索情報に基づいて検出し、かつ各検出処理要求元に係るメッセージ・キューに前記メッセージを挿入するスレッド設定ステップ、

各スレッドにそれが処理を担当するエージェント起動原因イベントを割当てる割当てステップ、

前記受付ステップにおいて受付けた各エージェント起動原因イベントについての各スレッドによる処理の進行状態情報を管理する進行状態情報管理ステップ、

処理進行状態情報がスレッド処理終了に係る情報になっているエージェント起動原因イベント（以下、「被判定エージェント起動原因イベント」と言う。）について、該被判定エージェント起動原因イベントより前に前記受付ステップにおいて受付けたエージェント起動原因イベントの中にまだスレッド処理の未終了になっているエージェント起動原因イベントが有るか無いかを判定する判定ステップ、及び

前記判定ステップにおいて「有る」と判定された被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェントによる処理を抑制するエージェント制御ステップ、

を有していることを特徴とするメッセージ処理方法。

【請求項２８】　前記判定ステップにおいて「無い」と判定された被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェントによる処理を許容する前記エージェント制御ステップ、

を有していることを特徴とする請求項２７記載のメッセージ処理方法。

【請求項２９】　「無い」と判定されたエージェント起動原因イベントに対して受付順番がその次のエージェント起動原因イベントが、「有る」との判定済

みのものであるときは、判定結果を「有る」から「無い」へ変更する前記判定ステップ、

を有していることを特徴とする請求項２７記載のメッセージ処理方法。

【請求項３０】　前記判定ステップによる判定結果が「無い」となっている複数個のエージェント起動原因イベントを生成起因とするメッセージが受付順番方向へ連続するメッセージ･キューについての処理をエージェントに実行させる場合、該エージェントにはそれら連続する複数個のメッセージについての処理を連続的に行わせるように前記エージェントを設定する前記エージェント設定ステップ、

を有していることを特徴とする請求項２９記載のメッセージ処理方法。

【請求項３１】　請求項１６～３０のいずれかのメッセージ処理方法における各ステップをコンピュータに実行させることを特徴とするメッセージ処理プログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、多数、例えば数十万～数百万又はそれ以上の処理要求元に適用するメッセージの処理を、エージェントを使って処理するメッセージ処理装置、メッセージ処理方法及びメッセージ処理プログラムに関し、詳しくは処理時間を短縮したメッセージ処理装置、メッセージ処理方法及びメッセージ処理プログラムに関する。

【０００２】

【従来の技術】

特許文献１では、エージェント・サーバが開示されている。該エージェント・サーバでは、負荷の増大を抑制するために、活動エージェントの個数を制限するとともに、エージェントを使って、各利用者へ所定のメッセージを通知することを開示する。

【０００３】

本出願人は非特許文献１に係るメッセージ処理サーバを出荷している。

【０００４】

【特許文献１】

特開平１１－３２７９０８号公報

【非特許文献１】

Agent Framework-Java（本出願人に係る製品の名称。http://www.alphaworks.ibm.com/aw_nsf/techs/caribbean閲覧。該ｗｅｂ頁では、該製品が出願時点においてコード・ネームcaribbeanとして紹介されている。）

【０００５】

【発明が解決しようとする課題】

メッセージ処理サーバの具体例に、所定の起因情報の受付けに対して会員へメール通知を行うメール配信サーバがある。このようなメール配信サーバにおける具体的ニーズについて考える。例えば、会員としての会員が自分の希望する通知（例：ＩＢＭの株価がＰ円以上に上がったことの通知。）の通知条件をサブスクライブ・テーブルに予め登録しておき、メッセージ処理サーバが、クライアント（例：所定の証券会社に設置されて企業の株価をオンラインで受け取っているコンピュータ）から株価変動等のｘ１を受けると、該ｘ１を起因とするメッセージの会員をサブスクライブ・テーブルから検索により特定し、特定された各会員へ通知するニーズが予想される。

【０００６】

特許文献１のエージェント・サーバ及び非特許文献１のメッセージ処理サーバは、各通知希望者の通知条件を予めサブスクライブ・テーブルに登録しておくこと、及びｘ１に対してサブスクライブ・テーブルから今回の通知対象の会員を検索して、会員へ通知メールを配信すると言う処理は行われていない。

【０００７】

メール配信サーバでは、次のニーズも予想される。すなわち、異なるｘ１が時間的に接近して生じ、異なるｘ１に対してそれぞれのｘ１に係る各通知メールを同一の会員へ配信する場合に、各通知メールに優先度を設定して、優先度順にメッセージへ送りたいことがある。例えば、典型的な通知希望者は、株価情報に係る通知メールは、他の内容の通知メール、例えば所定分野の新商品案内の通知メ

ールよりも早く受け取ることを希望する。サーバに実装される通知メール用メッセージ処理アプリケーションは、全体の処理時間短縮のために、シングル・スレッドではなく、マルチ・スレッドを利用するので、各ｘ１に対してスレッドを割当て、各会員に係るメッセージ・キューにメッセージを挿入することにすると、会員総数の多いメッセージが各会員用のメッセージ・キューに挿入されるのに要する時間が、会員総数の少ないメッセージが各会員用のメッセージ・キューに挿入されるのに要する時間より長くなり、優先度の高い通知メールが優先度の低い通知メールより会員に遅れて届くことがあり得る。

【０００８】

また、会員の格付けとしてゴールド会員と一般会員とが含まれている場合に、同一情報に係る通知メールはゴールド会員に一般会員より先に、すなわちゴールド会員の優先度を一般会員のそれよりも高くするのが営業政策上、望まれる。また、一方では、格付けの高い会員への優先度を高くしつつ、他方では、格付けの低い会員に対しても、急ぎの、すなわち内容的に優先度の高い通知メールについては、格付けの高い会員への急がない通知メールよりも早く（優先して）配信したいニーズも予想される。

【０００９】

別の予想ニーズとして、同一の会員へ配布しかつ内容の異なる複数の通知メールが存在するときは、それら通知メールは、それらの起因になる情報の受付順に配布しなければならないことがある。例えば、ＩＢＭの株価がＰ１になり、その数秒後にＰ２（Ｐ２＜Ｐ１）になった場合には、それら両方のｘ１を起因とする通知メールについて通知希望を出している通知希望者は、Ｐ１に係る通知メールを先に受けてから、Ｐ２に係る通知メールを受けないと、株価の値下がりにもかかわらず、株価の値上がりと誤解する。マルチ・スレッド型メッセージ処理アプリケーションを開発した場合、該アプリケーションでは、複数のスレッドが並列に処理を行うことになるので、各スレッドの処理量の多寡によっては、後のｘ１に割当てられたスレッドが、先のｘ１に割当てられたスレッドより先に処理を終了してしまい、通知希望者には、時間的に後のｘ１に係る通知メールが、時間的に前のｘ１に係る通知メールより先に届いてしまうことがあり得る。

【００１０】

本発明の目的は、エージェント起動原因イベントに対して対応の処理要求元に適用するメッセージについて、対応のエージェントに所定の処理を行わせるメッセージ処理装置、メッセージ処理方法及びメッセージ処理プログラムにおいて、作業時間の短縮を図ることである。

【００１１】

本発明の他の目的は、永続記憶装置からキャッシュ・メモリへのエージェントの読込みを抑制して、全体の処理時間の短縮を図りつつ、優先度に基づく処理を達成するメッセージ処理装置、メッセージ処理方法及びメッセージ処理プログラムを提供することである。

【００１２】

本発明の別の目的は、マルチ・スレッドによる処理を実現しつつ、各処理要求元に適用されるメッセージを、それらの起因となったエージェント起動原因イベントの受付け順に処理することを保障するメッセージ処理装置、メッセージ処理方法及びメッセージ処理プログラムを提供することである。

【００１３】

【課題を解決するための手段】

本発明のメッセージ処理装置は次のものを有する。

エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能でとなる複数個のエージェ

ント、

前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込み手段、

メッセージが挿入されているメッセージ・キューに係るエージェントにその作動を指示するエージェント用指示手段、及び

前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込み手段にその処理の繰返しを指示する繰返し指示手段。

【００１４】

エージェントを作動させるためには、エージェントがキャッシュ・メモリに存在する必要がある。処理要求元、したがってエージェントは、例えば、数十万から数百万、又はそれ以上に達することがあり、全部のエージェントをキャッシュ・メモリに常時、存在させるキャッシュ・メモリの容量上、困難である。本発明では、エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を処理要求元検索情報に基づいて作成し、該リスト情報に未選択の処理要求元として残っている全部（未選択の処理要求元が少ないとき）又は幾つかの（未選択の処理要求元が多いとき）を選択し、それら選択した処理要求元に対応付けられているメッセージ・キューにメッセージを挿入する。挿入・読込み手段は、該挿入作業に並行して又は挿入作業の前若しくは後に、それら処理要求元に対応付けられているエージェントを永続記憶装置からキャッシュ・メモリへ読込む。こうして、メッセージが挿入されているメッセージ・キューについては、該メッセージ・キューに係るエージェントは例えばスケジューラ等のエージェント用指示手段からの指示により作動開始することになるが、作動対象のエージェントはキャッシュ・メモリにおける存在を保証されるので、永続記憶装置からキャッシュ・メモリへのエージェントの読込みが抑制され、作業時間が短縮される。

【００１５】

エージェント起動原因イベントとしては、例えば株価の変更、電子モール・システムにおける商品データベースの価格変更等がある。各エージェントは、例えばＩＢＭの株価に係るエージェント起動原因イベントと言うように、起動原因が同種のものであるとは限定されない。例えば、同一のエージェントが、パソコンの価格に係るデータベースの更新としてのエージェント起動原因イベントと、プリンタの価格に係るデータベースの更新としてエージェント起動原因イベントとを起動原因とするように、複数主のエージェント起動原因イベントを起動原因とすることもあり得る。本発明のメッセージ処理装置は、メール配信処理装置に限定されない。同一のエージェントは、或るエージェント起動原因イベント（該エージェント起動原因イベントをエージェント起動原因イベントＡ１と呼ぶことにする。）に対しては、処理要求元へメールを通知することがあっても、別の或るエージェント起動原因イベント（該エージェント起動原因イベントをエージェント起動原因イベントＡ２と呼ぶことにする。）に対しては処理要求元へメールを通知しないこともある。また、エージェント起動原因イベントは、同一のエージェント起動原因イベント（該エージェント起動原因イベントをエージェント起動原因イベントＡ３と呼ぶことにする。）に対して、エージェント起動原因イベントＡ３が発生する前に起動原因となった１個又は複数のエージェント起動原因イベントの相違及びそれらの組み合わせの相違により、異なる処理結果、例えば対応の処理要求元へメールを通知したり、通知しなかったりすることもある。なお、本発明のメッセージ処理装置を、メール配信装置以外に適用したものは、後述の実施例４において詳述する。

【００１６】

処理要求元は、本発明のメッセージ処理装置、メッセージ処理方法及びメッセージ処理プログラムに対して外部から処理を要求するものである。処理要求元の下位概念には、例えば会員（有料会員及び無料会員を含む。また、会員には、人だけでなく、会社等の組織も会員の概念に含む。）、処理依頼元、アプリケーション処理要求元、処理委託元、サーバ処理要求元がある。処理要求元には、ワーク・フロー処理から処理要求を出すワーク・フロー処理部分も含まれる。これに

ついては、後述の実施例５において詳述する。

【００１７】

エージェントは、典型的には、メッセージ・キュー内のメッセージを引数として所定の処理を実施するメッセージ・ハンドラと、該メッセージ・ハンドラが使用するデータとを含む。例えば、エージェント起動原因イベントＡを生成起因とするメッセージをＢとすると、メッセージを適用される全処理要求元に対応付けられている各メッセージ・キューには、すべてメッセージＢが挿入されるが、エージェント内の処理結果はエージェント内のデータによってメッセージＢに対する処理結果が処理要求元ごとに異なることがある。例えば、メッセージ配信処理装置では、企業１の株価が＄９０から＄１００になったとき、処理要求元１，２は、＄１００以上になったとの通知メールを受け、処理要求元３は、＄９５以上になったとの通知メールを受ける。

【００１８】

本発明のメッセージ処理装置は次のものを有している。

エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能でとなる複数個のエージェント、

第１のメッセージ・キュー用処理機構、

第２のメッセージ・キュー用処理機構、

第１及び第２のメッセージ・キュー用処理機構のどちらかを選択する選択手段

、及び

メッセージが挿入されているメッセージ・キューに係るエージェントについて該エージェントがキャッシュ・メモリに有れば該エージェントにその作動を直ちに指示しまた該エージェントがキャッシュ・メモリに無ければ該エージェントを前記永続記憶装置から前記キャッシュ・メモリへ読込んでから該エージェントにその作動を指示するエージェント用指示手段。

さらに、第１のメッセージ・キュー用処理機構は、前記リスト情報に含まれる全部の処理要求元に係るメッセージ・キューに前記メッセージを挿入する挿入手段、を有している。また、第２のメッセージ・キュー用処理機構は、前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込み手段、及び前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込み手段にその処理の繰返しを指示する繰返し指示手段、を有している、

【００１９】

各エージェント起動原因イベントに係る処理要求元の総数は各エージェント起動原因イベントごとに異なる。非常に多い場合もあれば、きわめて少ない場合もある。また、各処理要求元に対応付けられている各エージェントがキャッシュ・メモリにおいてヒットする率は、状況によって異なる。類似するエージェント起動原因イベントが連続的に受付けられるときは、各エージェント起動原因イベントに係る処理要求元が一致する可能性が高まるので、各エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元に対応付けられるエージェントについてのキャッシュ・メモリにおけるヒット率が上昇する。そのような場合は、時として、（ａ）リスト情報に含まれる処理要求元の中から未選択の処理要求元を選択して、それら選択された処理要求元に対応付けられるメッセージ・キューの全部にメッセージを挿入し、かつそれら全部のメッセージ・キューに対応付けられるエージェントをキャッシュ・メモリへ読込むよりは、（ｂ）

作動させようとするエージェントがキャッシュ・メモリに有るときは該エージェントを直ちに、また、無いときは、該エージェントを永続記憶装置からキャッシュ・メモリへ読込んでから、作動させた方が作業時間が短縮されることもあり得る。本発明では、（ａ）と（ｂ）とを選択自在にする。（ａ）と（ｂ）との切替は、例えば、エージェント起動原因イベントごと、曜日ごと、季節ごと、及び時間帯ごとに実施できる。

【００２０】

本発明のメッセージ処理装置は次のものを有している。

エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元を前記処理要求元検索情報に基づいて決定する処理要求元決定手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているとき作動可能となる複数個のエージェント、

各メッセージについての処理優先度を単一の価値基準に基づいてサブ処理優先度として決定する少なくとも１個のサブ処理優先度決定手段、

前記サブ処理優先度決定手段の総個数が２以上であるときは各サブ処理優先度決定手段が各メッセージについて個々に決定したサブ処理優先度に基づいて各メッセージについての複合処理優先度を決定し、また、前記サブ処理優先度決定手段の総個数が１であるときは該唯一のサブ処理優先度決定手段が各メッセージについて決定したサブ処理優先度を複合処理優先度として決定する複合処理優先度決定手段、

各メッセージ・キューが保持しているメッセージの中で最高複合処理優先度のメッセージを最優先メッセージとし該最優先メッセージの複合処理優先度が同一であるメッセージ・キューに係るエージェント同士では、キャッシュ・メモリに存在する方のエージェントを、存在しない方のエージェントより優先してその作

動を指示するエージェント用指示手段。

【００２１】

処理要求元処理装置全体の処理時間を抑制しつつ、メッセージについて処理優先度を設定し、該処理優先度に基づいたメッセージの処理を行うニーズが存在する。処理優先度は、また、単一の価値基準により決定される場合もあるし、複数個の価値基準の処理優先度を複合して決定される場合もある。本発明では、複合処理優先度に基づき処理が行われるとともに、メッセージ・キュー間で、最高複合処理優先度が同一のメッセージが含まれる場合には、該メッセージ・キューに対応付けられるエージェントがキャッシュ・メモリに存在する方のメッセージ・キューを、存在しない方のメッセージ・キューより優先して、処理する。こうして、今回のエージェント起動原因イベントに対して作動対象となっているエージェントが、キャッシュ・メモリに存在していたときに作動時期が来ずに、作動を先延ばしされて、該エージェントがキャッシュ・メモリに存在しなくなってから、配布順番が来て、永続記憶装置からキャッシュ・メモリへ該エージェントを読込まなければならない事態が排除される。

【００２２】

本発明のメッセージ処理装置は次のものを有している。

エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記受付手段が受付けたエージェント起動原因イベントについての受付順番情報を管理する受付順番情報管理手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能となっている複数個のエージェント、

相互に並列動作可能である複数個のスレッドであって各スレッドはエージェン

ト起動原因イベントを起因とするメッセージが適用される処理要求元を処理要求元検索情報に基づいて検出しかつ各検出処理要求元に係るメッセージ・キューに前記メッセージを挿入する複数個のスレッド、

各スレッドにそれが処理を担当するエージェント起動原因イベントを割当てる割当て手段、

前記受付手段が受付けた各エージェント起動原因イベントについてのスレッドによる処理の進行状態情報を管理する進行状態情報管理手段、

処理進行状態情報がスレッド処理終了に係る情報になっているエージェント起動原因イベント（以下、「被判定エージェント起動原因イベント」と言う。）について、該被判定エージェント起動原因イベントより前に前記受付手段において受付けたエージェント起動原因イベントの中にまだスレッド処理の未終了になっているエージェント起動原因イベントが有るか無いかを判定する判定手段、及び

前記判定手段が「有る」と判定した被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェントによる処理を抑制するエージェント制御手段。

【００２３】

エージェント起動原因イベントは次々に受付けられ、また、各エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元の個数は膨大である。したがって、各エージェント起動原因イベントに対して、該エージェント起動原因イベントに係る処理要求元のメッセージ・キューへ、該エージェント起動原因イベントを生成起因とするメッセージを挿入する処理は、別々のスレッドに実施させる、すなわち複数のスレッドにより処理することが、作業時間短縮上、好ましい。しかし、マルチ・スレッドを採用する場合、各エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元の総数によっては、受付順番が後のエージェント起動原因イベントに係る処理を担当したスレッドが、受付順番が前のエージェント起動原因イベントに係る処理を担当したスレッドより先に処理を終了することがある。本発明では、メッセージ・キューにおけるメッセージの挿入処理において、受付順番が後のエージェント起動原因イベントに係る処理が終了しても、受付順番が前のエージェント起動原因イベン

トに係る処理が終了していなければ、後のエージェント起動原因イベントを生成起因とするメッセージについての処理要求元へのエージェントによる処理を抑制する。

【００２４】

本発明のメッセージ処理方法は次のステップを有している。

エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能であり各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能であるエージェント設定ステップ、

前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込みステップ、

メッセージが挿入されているメッセージ・キューに係るエージェントにその作動を指示するエージェント用指示ステップ、及び

前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込みステップの繰返しを指示する繰返し指示ステップ。

【００２５】

本発明の別のメッセージ処理方法は次のステップを有している。

エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能であり各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能であるエージェント設定ステップ、

第１のメッセージ・キュー用処理ステップ、

第２のメッセージ・キュー用処理ステップ、

第１及び第２のメッセージ・キュー用処理ステップのどちらかを選択する選択ステップ、及び

メッセージが挿入されているメッセージ・キューに係るエージェントについて該エージェントがキャッシュ・メモリに有れば該エージェントにその作動を直ちに指示しまた該エージェントがキャッシュ・メモリに無ければ該エージェントを前記永続記憶装置から前記キャッシュ・メモリへ読込んでから該エージェントにその作動を指示するエージェント用指示ステップ。

さらに、第１のメッセージ・キュー用処理ステップは、前記リスト情報に含まれる全部の処理要求元に係るメッセージ・キューに前記メッセージを挿入する挿入ステップ、を有している。また、第２のメッセージ・キュー用処理ステップは、前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込みステップ、及び前記リスト情報に含まれる処理要求元の中に未選択のものが残

っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込みステップの繰返しを指示する繰返し指示ステップ、を有している、

【００２６】

本発明の別のメッセージ処理方法は次のステップを有している。

エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元を前記処理要求元検索情報に基づいて決定する処理要求元決定ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは各処理要求元に対応付けられており永続記憶装置に記憶され各エージェントは永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能でありかつ各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能となっているように設定するエージェント設定ステップ、

各メッセージについての処理優先度を単一の価値基準に基づいてサブ処理優先度として決定する少なくとも１個のサブ処理優先度決定ステップ、

前記サブ処理優先度決定ステップの総個数が２以上であるときは各サブ処理優先度決定ステップにおいて各メッセージについて個々に決定したサブ処理優先度に基づいて各メッセージについての複合処理優先度を決定し、また、前記サブ処理優先度決定ステップの総個数が１であるときは該唯一のサブ処理優先度決定ステップにおいて各メッセージについて決定したサブ処理優先度を複合処理優先度として決定する複合処理優先度決定ステップ、及び

各メッセージ・キューが保持しているメッセージの中で最高複合処理優先度のメッセージを最優先メッセージとし該最優先メッセージの複合処理優先度が同一であるメッセージ・キューに係るエージェント同士では、キャッシュ・メモリに存在する方のエージェントを、存在しない方のエージェントより優先してその作

動を指示するエージェント用指示ステップ。

【００２７】

本発明の別のメッセージ処理方法は次のステップを有している。

エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記受付ステップにおいて受付けたエージェント起動原因イベントについての受付順番情報を管理する受付順番情報管理ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは、各処理要求元に対応付けられており、永続記憶装置に記憶され、永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能であり、かつ各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能となっているように設定するエージェント設定ステップ、

複数個のスレッドを設定するスレッド設定ステップであって各スレッドは、相互に並列動作可能であり、エージェント起動原因イベントを起因とするメッセージが適用される処理要求元を処理要求元検索情報に基づいて検出し、かつ各検出処理要求元に係るメッセージ・キューに前記メッセージを挿入するスレッド設定ステップ、

各スレッドにそれが処理を担当するエージェント起動原因イベントを割当てる割当てステップ、

前記受付ステップにおいて受付けた各エージェント起動原因イベントについての各スレッドによる処理の進行状態情報を管理する進行状態情報管理ステップ、

処理進行状態情報がスレッド処理終了に係る情報になっているエージェント起動原因イベント（以下、「被判定エージェント起動原因イベント」と言う。）について、該被判定エージェント起動原因イベントより前に前記受付ステップにおいて受付けたエージェント起動原因イベントの中にまだスレッド処理の未終了になっているエージェント起動原因イベントが有るか無いかを判定する判定ステッ

プ、及び

前記判定ステップにおいて「有る」と判定された被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェントによる処理を抑制するエージェント制御ステップ。

【００２８】

本発明のメッセージ処理プログラムは、前記各メッセージ処理処理方法における各ステップをコンピュータに実行させる。

【００２９】

【発明の実施の形態】

本発明の実施の形態について説明する。なお、発明の実施の形態及びその後に続いて説明する実施例は、本発明をそれらに限定するものではなく、その要旨を逸脱しない範囲で種々変更可能であることは言うまでもない。

【００３０】

図１はメッセージ処理システム１０の概略図である。ネットワーク１２は典型的にはインターネットである。本発明に従うアプリケーションを実装するメッセージ処理サーバ１４、複数個の情報源クライアント１５、及び例えば数十万～数百万又はそれ以上の処理要求元コンピュータ１６は、ネットワーク１２を介してメッセージ、データ、及び各種情報を相互に送受自在になっている。なお、ワークフロー内の所定部分が処理要求元になる場合、１個のコンピュータ１６が相互に識別自在の複数個の処理要求元になることがあり得る。情報源クライアント１５は、例えば、エージェント起動原因イベントとしての株価情報をメッセージ処理サーバ１４へ送るために、証券会社等に設置される。処理要求元コンピュータ１６は、ネットワーク１２へ直接接続されていることもあるし、ルータを介してネットワーク１２へ接続されていることもある。

【００３１】

図２はメッセージ処理装置２０の構成図である。メッセージ処理装置２０において、処理要求元検索情報管理手段２２は、エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報２１を管理する。処理要求元検索情報２１は、ハード・ディスク装置や磁気テープ装置等の永続記憶装

置に記憶される。受付手段２７はエージェント起動原因イベントを受付ける。リスト情報作成手段２８は、エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を処理要求元検索情報２１に基づいて作成する。複数個のエージェント２３は、各処理要求元に対応付けられており、永続記憶装置２４に記憶され、永続記憶装置２４からプログラム実行領域としてのキャッシュ・メモリ２５へ読込み可能及びキャッシュ・メモリ２５から破棄可能となっている。各エージェント２３は、キャッシュ・メモリ２５に存在しているときのみ、作動して、該エージェント２３に対応のメッセージ・キュー内のメッセージについての処理を実施可能でとなる。挿入・読込み手段３８は、リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し、該挿入・読込み対象処理要求元に係るメッセージ・キュー３９に、メッセージを挿入するとともに、挿入・読込み対象処理要求元に係るエージェント２３を永続記憶装置２４からキャッシュ・メモリ２５へ読込む、エージェント用指示手段３１は、メッセージが挿入されているメッセージ・キュー３９に係るエージェント２３にその作動を指示する。繰返し指示手段３２は、リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェント２３の処理終了を待って挿入・読込み手段３８にその処理の繰返しを指示する。

【００３２】

エージェント２３が、永続記憶装置２４からキャッシュ・メモリ２５へ読込まれることを「スワップ・インＳｉ」、及び永続記憶装置２４において上書き等で消去することを「スワップ・アウトＳｏ」と適宜、呼ぶことにする。スワップ・インＳｉ及びスワップ・アウトＳｏは、１個のエージェント２３を単位にして可能である。なお、スワップ・インＳｉ及びスワップ・アウトＳｏは、図２及び図４等では、幅太の矢印で示され、図５では、幅太の矢印と共に、幅細の矢印でも示されている。幅太の矢印によるスワップ・インＳｉ及びスワップ・アウトＳｏは、複数のエージェント２３をまとめてスワップ・インＳｉ及びスワップ・アウトＳｏすることを表し、幅細の矢印によるスワップ・インＳｉ及びスワップ・アウトＳｏは１個のエージェント２３をスワップ・インＳｉ及びスワップ・アウト

Ｓｏすることを表している。

【００３３】

メッセージ処理装置２０は例えばメール配信装置として使用される場合には、各エージェント２３は、該エージェント２３に対応付けられているメッセージ・キュー３９におけるメッセージに対して、該エージェント２３に対応付けられた処理要求元への通知を実施する。

【００３４】

なお、メール配信装置において、エージェントの処理終了とは、典型的には、エージェントが、その処理要求元のメール・アドレスを管理する処理要求元側メール・サーバへの配布終了である。しかし、処理要求元のメール・アドレスが、過去は存在していたものの、現在では、処理要求元側メール・サーバから消失していたり、処理要求元側サーバの一時的なトラブル等により配布でなかったり等の配布上の支障がある場合には、エージェントは、一時的又は永続的に処理が困難になることがある。このような場合には、エージェントは直ちに又は所定条件が満たされるや、処理を終了し、この終了を「エージェントの処理終了」とすることができる。所定条件とは、例えば一定時間が経っても配布上の支障が解決されないこととか、相手側メール・サーバへの配布を所定回数、試してみたが、失敗に終わったこととかである。

【００３５】

図３はエージェント２３の構成図である。エージェント２３は、メッセージを引数として渡されて所定の処理を実施するメッセージ・ハンドラ３４と、該メッセージ・ハンドラ３４がその処理の際に使用するデータ３５とを含む。

【００３６】

図４はメッセージ処理装置４２の構成図である。メッセージ処理装置４２において、図２のメッセージ処理装置２０の要素と同一のものは同符号で指示して、説明は省略する。選択手段４３は、第１及び第２のメッセージ・キュー用処理機構４５のどちらかを選択する。エージェント用指示手段４６は、メッセージが挿入されているメッセージ・キュー３９に係るエージェント２３について、該エージェント２３がキャッシュ・メモリ２５に有れば、該エージェント２３にその作

動を直ちに指示し、また、該エージェント２３がキャッシュ・メモリ２５に無ければ、該エージェント２３を永続記憶装置２４からキャッシュ・メモリ２５へ読込んでから、該エージェント２３にその作動を指示する。

【００３７】

図５及び図６はそれぞれ第１のメッセージ・キュー用処理機構４４及び第２のメッセージ・キュー用処理機構４５の詳細図である。第１のメッセージ・キュー用処理機構４４は、リスト情報に含まれる全部の処理要求元に係るメッセージ・キュー３９にメッセージを挿入する。第２のメッセージ・キュー用処理機構４５は、挿入・読込み手段３８及び繰返し指示手段３２を含む。第２のメッセージ・キュー用処理機構４５の挿入・読込み手段３８及び繰返し指示手段３２は図２のメッセージ処理装置２０の挿入・読込み手段３８及び繰返し指示手段３２と同一であり、説明を省略する。

【００３８】

図４に戻って、選択手段４３における選択はオペレータの指示に基づく。選択手段４３選択はオペレータの指示に基づく。又は、選択手段４３における選択は、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元の予測数、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元に係るエージェント２３についてのキャッシュ・メモリ２５における予測ヒット率、選択手段４３が第１のメッセージ・キュー用処理機構４４を選択している場合に受付手段２７が情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェント２３のメッセージ・キュー３９に挿入するまでの予測作業時間、選択手段４３が第２のメッセージ・キュー用処理機構４５を選択している場合に受付手段２７が情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェント２３のメッセージ・キュー３９に挿入するまでの予測作業時間、キャッシュ・メモリ２５内のエージェント２３についてその使用を決定してから該決定したエージェント２３が処理完了するまでの予測時間、及び／又はキャッシュ・メモリ２５外のエージェント２３についてその使用を決定してから該決定したエージェント２３が処理完了するまでの

予測時間に基づく。

【００３９】

図７はメッセージ処理装置５２の構成図である。メッセージ処理装置５２における処理要求元検索情報２１、処理要求元検索情報管理手段２２、エージェント２３、永続記憶装置２４、キャッシュ・メモリ２５、受付手段２７及びメッセージ・キュー３９は、前述のメッセージ処理装置２０におけるそれらと同一であり、説明は省略する。処理要求元決定手段５３は、エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元を処理要求元検索情報２１に基づいて決定する。少なくとも１個のサブ処理優先度決定手段５４ａ．５４ｂ．・・・は、各メッセージについての処理優先度を単一の価値基準に基づいてサブ処理優先度として決定する。複合処理優先度決定手段５５は、サブ処理優先度決定手段５４ａ．５４ｂ．・・・の総個数が２以上であるときは各サブ処理優先度決定手段５４ａ．５４ｂ．・・・が各メッセージについて個々に決定したサブ処理優先度に基づいて各メッセージについての複合処理優先度を決定する。複合処理優先度決定手段５５は、また、サブ処理優先度決定手段の総個数が１であるときは該唯一のサブ処理優先度決定手段（例えば５４ａ）が各メッセージについて決定したサブ処理優先度を複合処理優先度として決定する。エージェント用指示手段５６は、各メッセージ・キューが保持しているメッセージの中で最高複合処理優先度のメッセージを最優先メッセージとし該最優先メッセージの複合処理優先度が同一であるメッセージ・キューに係るエージェント２３同士では、キャッシュ・メモリ２５に存在する方のエージェント２３を、存在しない方のエージェント２３より優先してその作動を指示する。

【００４０】

価値基準には、メッセージの内容に係るもの及びメッセージが適用される処理要求元に係るものがある。メッセージの内容に係る所定の価値基準には、メッセージの処理の緊急性に係る価値基準が含まれる。メッセージが適用される処理要求元に係る価値基準には、処理要求元の格付けに係る価値基準が含まれる。処理要求元として、例えばゴールド処理要求元、シルバー処理要求元及びブロンズ処理要求元が存在する場合、処理要求元の格付けはゴールド処理要求元、シルバー

処理要求元及びブロンズ処理要求元の順番になる。

【００４１】

エージェント用指示手段４６により処理を一旦、開始したエージェント２３は、該エージェント２３に係るメッセージ・キューが保持するメッセージが複数個あるとき、それら全部のメッセージについて、又は複合処理優先度が上位から所定番目までのメッセージについて連続処理する。

【００４２】

図８は図７のサブ処理優先度決定手段５４ａ，５４ｂ，・・・及び複合処理優先度決定手段５５を構成要素として含むエージェント管理手段５９の全体構成図である。エージェント管理手段５９は、サブ処理優先度決定手段５４ａ，５４ｂ，・・・及び複合処理優先度決定手段５５を含む。エージェント管理手段５９は、さらに、各エージェント２３が永続記憶装置２４に存在するか否かを検出する存在検出手段６０、該存在検出手段６０の判定結果及び各エージェント２３の複合処理優先度に基づいてエージェント２３をグループ化しグループ化情報を管理するグループ化情報管理手段６１、及びグループ化情報管理手段６１にグループ化情報の更新を指示する更新指示手段６２を有している。エージェント用指示手段５６は、エージェント管理手段５９におけるグループ化情報に基づく順番でエージェント２３にその作動を指示する。

【００４３】

図９はメッセージ処理装置６４の構成図である。メッセージ処理装置６４における処理要求元検索情報２１、処理要求元検索情報管理手段２２、エージェント２３、永続記憶装置２４、キャッシュ・メモリ２５、受付手段２７及びメッセージ・キュー３９は前述のメッセージ処理装置２０におけるそれらと同一であり、説明は省略し、相違点を中心に述べる。受付順番情報管理手段６５は、受付手段２７が受付けたエージェント起動原因イベントについての受付順番情報を管理する。複数個のスレッド６７ａ，６７ｂ，・・・は相互に並列動作可能である。これらスレッド６７ａ，６７ｂ，・・・はエージェント起動原因イベントを起因とするメッセージが適用される処理要求元を処理要求元検索情報２１に基づいて検出し、かつ各検出処理要求元に係るメッセージ・キュー３９にメッセージを挿入

する。割当て手段６６は、各スレッド６７ａ，６７ｂ，・・・にそれが処理を担当するエージェント起動原因イベントを割当てる。進行状態情報管理手段７０は、受付手段２７が受付けた各エージェント起動原因イベントについてのスレッド６７ａ，６７ｂ，・・・による処理の進行状態情報を管理する。判定手段７２は、処理進行状態情報がスレッド６７ａ，６７ｂ，・・・処理終了に係る情報になっているエージェント起動原因イベント（以下、「被判定エージェント起動原因イベント」と言う。）について、該被判定エージェント起動原因イベントより前に受付手段２７において受付けたエージェント起動原因イベントの中にまだスレッド６７ａ，６７ｂ，・・・処理の未終了になっているエージェント起動原因イベントが有るか無いかを判定する。エージェント制御手段７３は、判定手段７２が「有る」と判定した被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェント２３による処理を抑制する。エージェント制御手段７３は、また、判定手段７２が「無い」と判定した被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェント２３による処理を許容する。

【００４４】

判定手段７２は、また、「無い」と判定したエージェント起動原因イベントに対して受付順番がその次のエージェント起動原因イベントが、「有る」との判定済みのものであるときは、判定結果を「有る」から「無い」へ変更する。図９のエージェント２３は、判定手段２７による判定結果が「無い」となっている複数個のエージェント起動原因イベントを生成起因とするメッセージが受付順番方向へ連続するメッセージ･キュー３９についての処理を実行する場合は、それら連続する複数個のメッセージについての処理を連続的に行うように、仕組まれている。

【００４５】

図１０はメッセージ処理方法のフローチャートである。Ｓ１００（処理要求元検索情報管理ステップ）では、エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報２１を管理する。Ｓ１０４（受付ステップ）では、エージェント起動原因イベントを受付ける。Ｓ１０５（リスト情

報作成ステップ）では、エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を処理要求元検索情報２１に基づいて作成する。Ｓ１０１（エージェント設定ステップ）では、複数個のエージェント２３を設定する。該設定では、各エージェント２３は、各処理要求元に対応付けられており、永続記憶装置２４に記憶され、永続記憶装置２４からプログラム実行領域としてのキャッシュ・メモリ２５へ読込み可能及びキャッシュ・メモリ２５から破棄可能であり、各エージェント２３はキャッシュ・メモリ２５に存在しているときのみ作動して該エージェント２３に対応のメッセージ・キュー内のメッセージについての処理を実施可能である。Ｓ１０６（挿入・読込みステップ）は、図１１において口述する。Ｓ１０８（エージェント用指示ステップ）では、メッセージが挿入されているメッセージ・キュー３９に係るエージェント２３にその作動を指示する。Ｓ１０７（繰返し指示ステップ）では、リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェント２３の処理終了を待ってＳ１０６（挿入・読込みステップ）の繰返しを指示する。

【００４６】

図１１は図１０のフローチャートにおけるＳ１０６（挿入・読込みステップ）の詳細図である。Ｓ１０６はＳ１１０及びＳ１１１を含む。Ｓ１１０及びＳ１１１は、時間並列的に実行されてもよいし、時間直列的に実行されてもよい、時間直列的に実行される場合、Ｓ１１０及びＳ１１１の順番は任意である。Ｓ１１０では、リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し、該挿入・読込み対象処理要求元に係るメッセージ・キュー３９に、メッセージを挿入する。Ｓ１１１では、挿入・読込み対象処理要求元に係るエージェント２３を永続記憶装置２４からキャッシュ・メモリ２５へ読込む。

【００４７】

図１２は別のメッセージ処理方法のフローチャートである。Ｓ１００，Ｓ１０１，Ｓ１０４及びＳ１０５は、図１０のそれらと同一内容であり、それらの説明は省略する。Ｓ１１５（選択ステップ）では、Ｓ１１６（第１のメッセージ・キ

ユー用処理ステップ）及びＳ１１７（第２のメッセージ・キュー用処理ステップ）のいずれかを選択する。Ｓ１１８（エージェント用指示ステップ）では、メッセージが挿入されているメッセージ・キュー３９に係るエージェント２３について該エージェント２３がキャッシュ・メモリ２５に有れば該エージェント２３にその作動を直ちに指示し、また、該エージェント２３がキャッシュ・メモリ２５に無ければ該エージェント２３を永続記憶装置２４からキャッシュ・メモリ２５へ読込んでから、該エージェント２３にその作動を指示する。

【００４８】

図１３は図１２のＳ１１６（第１のメッセージ・キュー用処理ステップ）の具体的な処理を示している。Ｓ１１６（第１のメッセージ・キュー用処理ステップ）は、リスト情報に含まれる全部の処理要求元に係るメッセージ・キュー３９にメッセージを挿入するＳ１１９（挿入ステップ）を含む。

【００４９】

図１４は図１２のＳ１１７（第２のメッセージ・キュー用処理ステップ）の具体的な処理を示している。Ｓ１１７はＳ１０６及びＳ１０７を含む。これらＳ１０６及びＳ１０７は、図１０のＳ１０６及びＳ１０７と同一であるので、説明は省略する。

【００５０】

図１２に戻って、Ｓ１１５（選択ステップ）における選択はオペレータの指示に基づく。又は、Ｓ１１５（選択ステップ）における選択は、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元の予測数、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元に係るエージェント２３についてのキャッシュ・メモリ２５における予測ヒット率、Ｓ１１６（第１のメッセージ・キュー用処理ステップ）において処理を割当てられた場合にＳ１０４（受付ステップ）において情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェント２３のメッセージ・キュー３９に挿入するまでの予測作業時間、Ｓ１１７（第２のメッセージ・キュー用処理ステップ）において処理を割当てられた場合にＳ１０４（受付ステップ）において情報を受け取ってからメ

ッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェント２３のメッセージ・キュー３９に挿入するまでの予測作業時間、キャッシュ・メモリ２５内のエージェント２３についてその使用を決定してから該決定したエージェント２３が処理完了するまでの予測時間、及び／又はキャッシュ・メモリ２５外のエージェント２３についてその使用を決定してから該決定したエージェント２３が処理完了するまでの予測時間に基づく。

【００５１】

図１５はさらに別のメッセージ処理方法のフローチャートである。Ｓ１００、Ｓ１０１及びＳ１０４は、図１０のそれらと同一であるので、説明は省略する。Ｓ１２５（処理要求元決定ステップ）では、エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元を処理要求元検索情報２１に基づいて決定する。少なくとも１個のＳ１２８ａ，ｂ，・・・（サブ処理優先度決定ステップ）では、各メッセージについての処理優先度を単一の価値基準に基づいてサブ処理優先度として決定する。Ｓ１２９（複合処理優先度決定ステップ）では、サブ処理優先度決定ステップの総個数が２以上であるときは各Ｓ１２８ａ，ｂ，・・・において各メッセージについて個々に決定したサブ処理優先度に基づいて各メッセージについての複合処理優先度を決定し、また、サブ処理優先度決定ステップの総個数が１であるときは該唯一のサブ処理優先度決定ステップ（例：Ｓ１２８ａ）において各メッセージについて決定したサブ処理優先度を複合処理優先度として決定する。Ｓ１３２（エージェント用指示ステップ）では、各メッセージ・キューが保持しているメッセージの中で最高複合処理優先度のメッセージを最優先メッセージとし該最優先メッセージの複合処理優先度が同一であるメッセージ・キューに係るエージェント２３同士では、キャッシュ・メモリ２５に存在する方のエージェント２３を、存在しない方のエージェント２３より優先してその作動を指示する。

【００５２】

Ｓ１２８ａ，Ｓ１２８ｂ，・・・（サブ処理優先度決定ステップ）において採用される価値基準には、メッセージの内容に係るもの及びメッセージが適用される処理要求元に係るものがある。さらに、メッセージの内容に係る所定の価値基

準には、メッセージの処理の緊急性に係る価値基準が含まれる。メッセージが適用される処理要求元に係る価値基準には、処理要求元の格付けに係る価値基準が含まれる。

【００５３】

Ｓ１３２（エージェント用指示ステップ）により処理を一旦、開始したエージェント２３は、該エージェント２３に係るメッセージ・キューが保持するメッセージが複数個あるとき、それら全部のメッセージについて、又は複合処理優先度が上位から所定番目までのメッセージについて連続処理する。

【００５４】

図１６は図１５のＳ１２８ａ，Ｓ１２８ｂ，・・・及びＳ１２９をサブステップとして含む上位ステップを備えるメッセージ処理方法部分のフローチャートである。Ｓ１３５（エージェント管理ステップ）は、Ｓ１２８ａ，Ｓ１２８ｂ，・・・及びＳ１２９の他に、Ｓ１３７及びＳ１３８を含む。Ｓ１３７（存在検出ステップ）では、各エージェント２３が永続記憶装置２４に存在するか否かを検出する。Ｓ１３８（グループ化情報管理ステップ）では、該Ｓ１３７（存在検出ステップ）の判定結果及び各エージェント２３の複合処理優先度に基づいてエージェント２３をグループ化しグループ化情報を管理するとともに、該グループ化情報を適宜更新する。Ｓ１３２（エージェント用指示ステップ）は、エージェント管理ステップにおけるグループ化情報に基づく順番でエージェント２３にその作動を指示する、

【００５５】

図１７はさらに別のメッセージ処理方法のメッセージ処理方法のフローチャートである。Ｓ１００、Ｓ１０１及びＳ１０４は図１０のそれらと同一であり、説明は省略する。Ｓ１４９（受付順番情報管理ステップ）では、Ｓ１０４（受付ステップ）において受付けたエージェント起動原因イベントについての受付順番情報を管理する。Ｓ１５０（スレッド設定ステップ）では、複数個のスレッド６７ａ，６７ｂ，・・・を設定する。この設定により、各スレッド６７ａ，６７ｂ，・・・は、相互に並列動作可能であり、エージェント起動原因イベントを起因とするメッセージが適用される処理要求元を処理要求元検索情報２１に基づいて検

出し、かつ各検出処理要求元に係るメッセージ・キュー３９にメッセージを挿入する。Ｓ１５２（割当てステップ）では、各スレッド６７ａ，６７ｂ，・・・にそれが処理を担当するエージェント起動原因イベントを割当てる。こうして、割当てられたスレッドはＳ１５３において作動開始する。Ｓ１５３（進行状態情報管理ステップ）では、Ｓ１０４（受付ステップ）において受付けた各エージェント起動原因イベントについての各スレッド６７ａ，６７ｂ，・・・による処理の進行状態情報を管理する。Ｓ１５６（判定ステップ）では、処理進行状態情報がスレッド６７ａ，６７ｂ，・・・処理終了に係る情報になっているエージェント起動原因イベント（以下、「被判定エージェント起動原因イベント」と言う。）について、該被判定エージェント起動原因イベントより前にＳ１０４（受付ステップ）において受付けたエージェント起動原因イベントの中にまだスレッド６７ａ，６７ｂ，・・・処理の未終了になっているエージェント起動原因イベントが有るか無いかを判定する。

【００５６】

図１８は図１７のＳ１５７（エージェント制御ステップ）の配信処理制御の具体例を示す。Ｓ１６０では、Ｓ１５６の判定結果に基づきＳ１６１又はＳ１６２へ分岐する。すなわち、Ｓ１５６の判定結果が「有る」の場合は、Ｓ１６１へ進み、また、Ｓ１５６の判定結果が「無い」の場合は、Ｓ１６２へ進む。Ｓ１６１では、Ｓ１５６（判定ステップ）において「有る」と判定された被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェント２３による処理を抑制する。Ｓ１６２では、Ｓ１５６（判定ステップ）において「無い」と判定された被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェント２３による処理を許容する。

【００５７】

好ましくは、判定ステップ１５６では、「無い」と判定されたエージェント起動原因イベントに対して受付順番がその次のエージェント起動原因イベントが、「有る」との判定済みのものであるときは、判定結果を「有る」から「無い」へ変更する。これにより、該次のエージェント起動原因イベントを起因とするメッセージは、エージェント２３による処理を速やかに開始される。さらに、Ｓ１０

１（エージェント設定ステップ）におけるエージェント２３の設定では、好ましくは、Ｓ１５６（判定ステップ）による判定結果が「無い」となっている複数個のエージェント起動原因イベントを生成起因とするメッセージが受付順番方向へ連続するメッセージ･キュー３９についての処理をエージェント２３に実行させる場合、該エージェント２３にはそれら連続する複数個のメッセージについての処理を連続的に行わせるように前記エージェントを設定する。

【００５８】

【実施例】

以下の実施例は、本発明をメッセージ処理装置の具体例としてのメール配信装置に適用したものである。

【００５９】

（実施例１）

実施例１の装置は、配信起因情報の受付に対して該配信起因情報に係る配信用メッセージを全配信先へ配信する際に、永続記憶装置からキャッシュ・メモリへのエージェントの読込みを能率的に行う方式として、KeyOnlyCollection（キー・オンリ・コレクション）配信機構又はSwapInCollection（スワップ・イン・コレクション）配信機構を装備する。該装置は、また、KeyOnlyCollection配信機構及びSwapInCollection配信機構の内、処理能率の良い方を状況に応じて選択する。実施例１の特徴を説明する。

【００６０】

［メッセージ配信サービス］

インターネットでは、利用者の要求を受けて結果を返す要求ー応答型のサービスだけでなく、利用者の嗜好や属性、要求情報をサーバに登録しておき、サーバ側で発生したイベントが発生すると、各利用者の情報を使って何らかの処理を行うイベント処理型のサービスが考えられる。例えば、株価情報の配信サービスでは、「ＩＢＭの株価が100ドル以上になったらメールで通知してくれ」というような要求が考えられる。このようなサービスでは、株の銘柄は、ある利用者では「ＩＢＭ」であり、他の利用者では「Ｍｉｃｒｏｓｏｆｔ社」である。また、株価通知の閾値（具体的株価。例：＄１００，＄１５０。）も利用者ごとに異なる

。例えば、同じＩＢＭの株価に興味のある人でも、別の利用者は株価が「90ドルを下回ったら通知してくれ」であるかもしれない。さらに、利用者によっては、「投資額の利益が1000ドル以上になったら通知してくれ」や「自分が前回購入した株価より10％値上がりしたら通知してくれ」ということもある。一方、インターネットの利用者数は数十万から数百万を超える場合がありえる。したがって、イベント処理型のサービスを提供するサイトでは、このような莫大な利用者に対してサービスを提供しなければならず、性能が非常に重要になる。

【００６１】

このようなサービスを実現するためのフレームワークと実行環境としてエージェント・サーバがある。このフレームワークでは、各利用者のエージェントをサーバに生成する。エージェントとは、利用者のデータとメッセージを処理する機構を持つオブジェクトである。サーバであるイベント（前述したメッセージ起因情報に対応する。）が発生すると、それを表すメッセージを生成し、そのメッセージに対してそれぞれのエージェントが処理を実行する。このとき、すべてのエージェントのメッセージ処理を実行するのではなく、そのメッセージに関連するエージェントだけの処理を実行することが性能向上のキーとなる。そのために、あらかじめ各エージェントは興味のあるメッセージの条件を登録しておき、エージェント・サーバは、その登録条件を参照して、必要なエージェントのメッセージ処理を行う。

【００６２】

１個のイベントが発生すると、そのメッセージを処理するために非常に多くのエージェントの処理を実行しなければならない。一方、このようなイベントが連続して発生する場合もある。したがって、エージェント・サーバは、イベント発生に伴う一連のエージェントの処理をなるべく効率よく実行することが大切である。

【００６３】

［エージェント・サーバ概要］

図１９はエージェント・サーバ２００の概略構成図である。各符号が指している要素は下記の通りである。

２０１：メッセージ

２０４：メッセージ受信部

２０５：エージェント・スケジューラ

２０６：スレッド・プール

２０７：スレッド・プール２０６に配置される複数個のスレッド

２１０：メッセージ・キュー用キャッシュ

２１１：メッセージ・キュー用キャッシュ２１０に配置される複数個のメッセージ・キュー

２１５：永続記憶領域（Persistent Area。例えばハード・ディスク装置や磁気テープ内に確保される。）

２１６：複数個のエージェント

２１８：エージェント・キャッシュ

【００６４】

図２０はエージェント・サーバ２００による種々の配信方法についての説明図である。各符号が指している要素は次の通りである。図１９の符号と同一のものについては説明を省略する。

２２０：クライアント

２２１：メッセージ・ブローカ

２２４：メッセージング機構

２２５：サブスクライブ・テーブル

【００６５】

図１９において、メッセージ２０１は、メッセージ・レシーバ２０１に受付けられ、配信先に対応するメッセージ・キュー２１１に挿入される。１個のメッセージ２０１に対する配信先の個数は典型的には何十万～何百万である。エージェント・スケジューラ２０５は、メッセージ挿入済みのメッセージ・キュー２１１に係るエージェント２１６について、所定のスケジュールに従って作動させる。エージェント２１６は、その作動に先立って、エージェント・キャッシュ２１８に呼び出されていなければならない。エージェント・サーバ２００全体の処理を効率化するために、全体の処理工程を複数個に分割し、各分割部分ごとにマルチ

・スレッド方式の処理が利用される。

【００６６】

図２０において、クライアント２２０は、前述の図１の情報源クライアント１５に相当し、株価等の情報源をもつコンピュータ内にアプリケーションとしてインストールされている。メッセージ・ブローカ２２１は、各クライアント２２０から各種の情報を受け取り、対応の情報をエージェント・サーバ２００へ流す。なお、メッセージ・ブローカ２２１とクライアント２２０及びエージェント・サーバ２００との間は、特に限定はしないが、インターネットを介して接続されている。メッセージング機構２２４は、メッセージ・ブローカ２２１より受信したメッセージとしての配信起因情報に対して、それに係る配信用メッセージについての全部の配信先を、サブスクライブ・テーブル２２５に基づいて検出する。１個の配信起因情報に対する配信先が１個である場合の配信を「ポイント・ツー・ポイント(Point-to-Point)」、また、１個の配信起因情報に対する配信先が複数個である場合の配信を「ファンアウト(FanOut)」と呼んでいる。

【００６７】

[エージェント・サーバ概要]

エージェント・サーバ２００は、個々のエージェントのデータを永続記憶領域２１５に保存し、そのデータから各エージェントをメモリ上に再構成する機構を持つ。永続記憶領域２１５にはディスクやデータベースなどが使われる。さらに、性能を向上させるために、エージェントをエージェント・サーバ２００のメモリ上にキャッシュする機構を持つ（以降、このキャッシュを「エージェント・キャッシュ２１８」と呼ぶ。）。エージェント・サーバ２００は、エージェント自体の管理と同時に、エージェントへのメッセージ２０１の配信も管理する。エージェント・サーバ２００は、外部から送信されてくるメッセージ２０１を受信して、そのエージェント・サーバ２００が管理するエージェントにそのメッセージ２０１を配信することを行う。

【００６８】

典型的なメッセージ２０１の配信方法として、１個のメッセージ２０１を特定のエージェントに配信するポイント・ツー・ポイント（P2P）と１個のメッセー

ジ２０１を複数のエージェントに配信するファンアウト（FanOut）がある。FanOutの場合は、各エージェントが事前に自分が受け取りたいメッセージ２０１の種類を登録しておき、エージェント・サーバ２００がFanOutを行う時に、その登録情報を参照して、配信すべきエージェントを選定するという方法（以降「Pub/Sub」と呼ぶ。）や、無条件にすべてのエージェントに配信する方法（以降「Multicast」と呼ぶ。）などがある。エージェント・サーバ２００はエージェントへのメッセージ２０１の配信を行うために、エージェントごとにメッセージ・キュー２１１を持っている。エージェント・サーバ２００は外部から送信されてくるメッセージ２０１を受信すると、そのメッセージ２０１を配信すべきエージェントを特定し、該当するエージェントのメッセージ・キュー２１１に受信したメッセージ２０１を格納する。

【００６９】

エージェント・サーバ２００はメッセージ・キュー２１１にメッセージ２０１があるエージェント２１６の中から適当なものを選択し、それにスレッド・プール２０６にあるスレッドを割当てる。スレッドは、それが割当てられたエージェント２１６のメッセージ・ハンドラ・ルーチン（以降「メッセージ・ハンドラ」と記述する）を呼び出す。このとき、スレッド２０７を割当てられエージェント２１６がエージェント・キャッシュ２１８にない場合、該エージェント２１６のデータを永続記憶領域２１５から読込み、エージェント２１６を再構成してエージェント・キャッシュ２１８に置く（このような処理を「スワップ・イン」と呼ぶ。）。エージェント・キャッシュ２１８が保持できるエージェント数には限界があるため、すでにエージェント・キャッシュ２１８がいっぱいの場合は、スレッド２０７が割当てられていないエージェント２１６をエージェント・キャッシュ２１８から破棄する（この処理を「スワップ・アウト」と呼ぶ。）。
エージェント２１６にスレッド２０７を割当てる時、どのエージェント２１６から先にスレッド２０７を割当てるかが性能に大きく影響する。これを行うのがエージェント・スケジューラ２０５である。エージェント・サーバ２００の典型的なアプリケーションとして、「株損益通知サービス」等の「通知サービス」がある。このようなアプリケーションでは、株価が更新されると、更新された株価に

興味のあるすべての利用者のエージェント２１６の処理を行わなければならない。一気に数万から数十万のエージェント２１６の処理を行う場合もある。性能を向上させるには、スワップ・イン及びスワップ・アウトの回数を減らすことが重要となる。なぜならば、スワップ・イン及びスワップ・アウトは永続記憶領域２１５へのアクセスを伴うものであり、通常のディスクやデータベースではこのコスト（時間上のコスト）は大変大きなものであるためである。エージェント・サーバ２００は、性能向上するための重要な３つの特徴を持っている。第１は、エージェント２１６は非同期メッセージ２０１を受けて処理を行うということである。ここで、「非同期」とは即座に処理が行われる必要がないということである。同期処理の場合は、そのメッセージ２０１の処理結果を待つプログラムが存在するため、これを受けたシステムはなるべく早く処理を行い、結果を返す必要がある。しかし、非同期処理では、即座に処理結果を返す必要がない。つまり、エージェント・サーバ２００はどのような順番でエージェント２１６の処理を行っても良いということである。第２は、エージェント・サーバ２００は各エージェント２１６のメッセージ・キュー２１１を持つということである。エージェント２１６に送られるメッセージ２０１は、いったんそのエージェント２１６のメッセージ・キュー２１１に入れられる。これにより、エージェント・サーバ２００は、どのエージェント２１６が処理待ち（メッセージ・キュー２１１にメッセージ２０１がある。）であり、どのエージェント２１６は処理を待っていないかを知ることができる。第３は、エージェント・サーバ２００はエージェント・キャッシュ２１８を管理しているということである。つまり、どのエージェント２１６がメモリ中にあり、どのエージェント２１６がメモリ中にないかを知っているのである。

【００７０】

［エージェント］

ここで言うエージェントとは、メッセージ処理を行うためのハンドラ・ルーチンを持ち、メッセージを受けると、そのエージェントが持つデータを用いて処理を遂行するオブジェクトである。エージェントは永続的であり、各エージェントが管理するデータはデータベース（データベースは永続記憶領域の概念に含まれ

る。）に保存されている。各エージェントには、それを識別するための「エージェント・キー」があり、エージェント・サーバは、エージェント・キーからエージェントを特定できる。

【００７１】

［エージェント・サーバとその管理機構］

エージェント・サーバは１個のプログラムであり、数十万から数百万を超えるエージェントの実行を管理するプログラムである。このプログラムはスレッド・プール機構を持ち、限られた数のスレッドを適宜エージェントに割当て、エージェントのメッセージ・ハンドラを呼び出す。また、このプログラムは、各エージェントに作られるメッセージ・キューを管理し、エージェントに配信されるメッセージを、一時的にこのキューに入れ、スレッドを割当てるときに、このキューからメッセージを取出し、メッセージ・ハンドラの引数として渡す。

【００７２】

エージェント・サーバは、エージェントをメモリ上に一時的に保持できるエージェント・キャッシュ機構を持つ。エージェント・サーバは、エージェントのメッセージ・ハンドラを呼び出す直前に、該当するエージェントのオブジェクトがキャッシュ上にある場合は、そのオブジェクトを利用する。しかし、キャッシュの大きさには限界があり、すべてのエージェントがキャッシュ上にあるとは限らない。エージェントのオブジェクトがキャッシュ上にない場合は、キャッシュ上にある他のエージェントのオブジェクトをキャッシュから破棄し（以降、この処理を「スワップ・アウト（Swap Out）」と呼ぶ。）、該当するエージェントのデータをデータベースからＳＱＬ（Structured Query Language）検索処理で読込み、キャッシュ上にエージェントのオブジェクトとして貼り付ける（以降、この処理を「スワップ・イン（Swap In）」と呼ぶ。）。

【００７３】

エージェント・サーバは、なるべくデータベースのアクセス回数を減らすために、エージェント・キャッシュ上にあるエージェントから優先的にスレッドを割当てる。スワップ・アウトさせる場合は、メッセージ・キューが空のエージェントから優先的にスワップ・アウトさせる。

【００７４】

［サブスクライブ］

エージェントは、自分が興味のあるメッセージの種類を示す情報をデータベースのテーブル（以降「サブスクライブ・テーブル」と呼ぶ。）に格納する。例えば、ＩＢＭ株価が100ドルを超えたことを示すメッセージを受信したい場合は、自分のエージェントを識別するためのエージェント・キー、株銘柄として「ＩＢＭ」、閾値として「100ドル」をデータベースのテーブルに格納する。或るメッセージがエージェント・サーバに到着すると、メッセージ配信機構は、そのメッセージの内容を参照し、サブスクライブ・テーブルからそのメッセージを配信すべきエージェントのキーのリストを取得し、それぞれのメッセージ・キューにメッセージを入れる。その後、エージェント・サーバは適切なスケジューリングにより、適宜エージェントのメッセージ・ハンドラを呼び出す。

【００７５】

［エージェント検索機構］

エージェント・サーバは、データベースに対して、検索条件を指定して、その条件に合致するエージェント・キーのリストを返す機構を提供する。エージェントのデータはデータベースのテーブルに保管されている。このテーブルにはエージェント・キーも含まれている。したがって、そのテーブルに対してＳＱＬ検索を行うことで、条件にあうエージェント・キーのリストを取得できる。この検索の方式として、下記の（ａ）及び（ｂ）の２方式を採用する。

【００７６】

（ａ）KeyOnlyCollection配信機構：

検索に合致したすべてのエージェント・キーをエージェント・サーバ内に読込み、リスト・オブジェクトを生成して呼出プログラムに返す方法。読込み方には、すべてのエージェント・キーを一気に読込む方法と順次読込む方法がある。後者の場合、呼出プログラムに戻されたリスト・オブジェクトから順次エージェント・キーを取得すると同時に、取得しようとしたエージェント・キーがまだメモリ中にない場合に、次のエージェント・キーを含む一塊のエージェント・キーの集団をデータベースからメモリ中に読込む。この方法では該当するエージェント

はスワップ・インされない。

（ｂ）SwapInCollection配信機構：

検索に合致したエージェント・キーのリストを表すリスト・オブジェクトを呼出プログラムに返す。この方法では、エージェント・キーを読込むと同時に、エージェントのデータも読込み、エージェントのオブジェクトをエージェント・キャッシュにスワップ・インする。エージェント・キーの読込み方は、KeyOnlyCollection配信機構の順次読込む方法と同様に行う。また、エージェントのデータベースからの読込みは、あらたなＳＱＬ検索を行うのではなく、リストを取得したＳＱＬ検索処理で読込む。

【００７７】

各方式（ａ）及び（ｂ）の特徴及び利点は次の通りである。

（ａ）KeyOnlyCollection配信機構：

この方式は、エージェント・キャッシュになんら影響を与えない。エージェント・サーバでは、エージェント・キャッシュの情報を利用して、なるべくスワップ・イン／アウトを少なくするようにエージェントの処理順序のスケジュールを行う。したがって、この方式では、エージェント・キャッシュの情報を最大限に利用してエージェントの処理のスケジュールを行うことができる。また、エージェント・キーのリストの取得のオーバーヘッドが小さい。したがって、この方式は、エージェント・キャッシュが大きくキャッシュ・ヒット率が高い場合か、エージェント・キャッシュのサイズは小さくても、検索結果に偏りがあり、エージェント・キーのリストのほとんどがエージェント・キャッシュにある場合に効果がある。

【００７８】

（ｂ）SwapInCollection配信機構：

この方式では、エージェント・キーのリストから適当な数のエージェント・キーを読込み、それらのメッセージ・キューにメッセージを入れ、エージェントがそのメッセージの処理を完了したら、次の適当な数のエージェント・キーをリストから取得し、同様の処理を行っていく。このとき、SwapInCollectionを利用しているため、エージェント・キーのリストからの読込みと同時に、そのキーに対

応したエージェントで、エージェント・キャッシュにないものがまとまってスワップ・インされる。この読込みは、１個のエージェントのスワップ・インのために１個のＳＱＬ検索処理を実行することを複数回繰り返す場合に比べ、非常に高速に複数のエージェントをスワップ・インすることができる。したがって、エージェント・キャッシュのサイズがエージェント・キーのリストの大きさに比べて小さく、キャッシュ・ヒット率が低い場合に効果がある。

【００７９】

なお、各エージェントの作動開始は、KeyOnlyCollection配信機構やSwapInCollection配信機構とは別の機構としてのスケジューラが管理しており、スケジューラは、作動開始させようとするエージェントがもしエージェント・キャッシュになければ、該エージェントを永続記憶領域からエージェント・キャッシュに読込むことになる。SwapInCollection配信機構が作動しているときは、配信用メッセージが挿入されたメッセージ・キューに係るエージェントはキャッシュ・メモリに読込まれていることが保障されるので、スケジューラは、作動開始させようとするエージェントを永続記憶領域からエージェント・キャッシュに読込む手間を省略される。

【００８０】

［メッセージの配信方法］

メッセージ配信機構は、メッセージをエージェント・キュー（＝メッセージ・キュー）に入れた後、或るエージェントがそのメッセージの処理を完了したことを知ることができる。一例として、メッセージ・リスナを利用する方法がある。メッセージ配信機構は、メッセージに対してメッセージ・リスナ・オブジェクト（以降「メッセージ・リスナ」と呼ぶ。）を関連付けることができる。エージェント・サーバは、或るエージェントがメッセージ処理を完了すると、そのメッセージに関連ついているメッセージ・リスナがあれば、そのコールバック・ルーチンを呼び出す。これにより、メッセージ配信機構は、メッセージ処理の完了を知ることができる。また、１個のメッセージを複数のエージェントに配信する場合、そのメッセージのメッセージ・リスナにカウンタを設けることで、すべてのエージェントがそのメッセージの処理を完了したかどうかを知ることができる。

【００８１】

この実装例として、WebSphere EJB+Programming Model Extensions (PME) を用いた場合を説明する。EJBではSQL文をツールに与えることで、検索を行うためのFinderメソッドを生成することができる。Finderメソッドは、その実装において与えられたSQL文を実行し、該当するレコードをラップしたEntityBeanインスタンスを示すプロキシであるEJBObjectインスタンスのCollectionを返す。このとき、PMEでは、Finderメソッドの戻り値のCollectionオブジェクトとしてKeyOnlyCollection配信機構とSwapInCollection配信機構のどちらにするかを、Finderメソッド生成時に指定することができる。

【００８２】

ここで、MessengerクラスとMessageListenerクラスを以下のように定義する。なお行番号を各行の先頭に挿入している。また"//"は注釈行を意味する。

Messengerクラス：

```
10:public class Messenger {
11:     javax.jms.Message msg;
12:     MessageListener listener;
13:     public Messenger(javax.jms.Message m, MessageListener lstnr) {
14://13行目ではmsgとlistenerにメッセージとしてのmと、リスナーとしてのlstnrをセット
15:     }
16:     public void postTo(AgentKey key){
17:
18://17行目には、 メッセージをKeyで指定されたエージェントのエージェント・キューに入れる処理を記述する。
19:     }
20:}
```

【００８３】

１１行及び１２行は、msg及びlistenerの変数宣言を行っている。１３～１５行ではメソッドMessengerを定義し、１６～１９行ではメソッドpostToを定義し

ている。m、lstnr、keyは各メソッドの引数である。

【００８４】

MessageListenerクラス：

```
30:public class MessageListener {
31:     int count = 0;
32:public MessageListener(int c) {
33:     //ここに、エージェント・キューにメッセージを入れた回数をセットする処理を記述する。
34:     }
35:     public void completed() {
36:         synchronized(this) {
37:             count--; // カウンタを１減らす
38:             if (count == 0) {
39:                 // もしすべてのエージェントがメッセージ処理を完了したら
40:                 // このMessageListenerオブジェクトでwaitしているスレッド
41:                 // を復帰させる。
42:             }
43:         }
44:     }
45:}
```

【００８５】

ここで、AgentKeyクラスはエージェントを特定するためのエージェント・キーのクラスであり、Finderメソッドの戻り値であるCollectionオブジェクトが持つEJBObjectインスタンスからAgentKeyインスタンスを取得できるとする。また、MessageListenerオブジェクトのcompletedメソッドは或るエージェントがメッセージの処理を完了するたびに呼び出される。もしMessengerオブジェクトのlistener変数がnullの場合は、エージェント・サーバの実行環境はcompletedメソッド

の呼び出しを行わない。

【００８６】

メッセージ配信機構は、上述のクラスを用いて以下のような処理を行う。KeyOnly配信機構では、今回の配信用メッセージを配信する全あて先リストに基づいて、各あて先に係る各メッセージ・キューに配信用メッセージが挿入される（６７行）。また、SwapIn配信機構では、今回の配信用メッセージを配信する全あて先リストに基づいて、該リストから例えば２０個ずつあて先を取出して（７５～８０行）、それら取出したあて先に係るエージェントをエージェント・キャッシュに読込み（８１～８２行）、それら取出した各あて先に係る各メッセージ・キューに配信用メッセージを挿入し（８６～８９行）、その後、それら取出した全あて先に係る全エージェントがその処理を終了するのを待つ（９０～９３行）。

【００８７】

[KeyOnly配信機構]

```
60:javax.jms.Message msg = // JMS providerから受けたメッセージをセット
61:KeyOnlyCollection collection = // Finderメソッドで検索を行う。
62:Messenger msgr = new Messenger(msg, null);
63:Iterator it = collection.iterator();
64:while(it.hasNext()) { // 要素があるまで繰り返す
65:     EJBObject obj = (EJBObject)it.next(); // EJBObjectを取出す
66:     AgentKey key = // objからエージェント・キーを取得する。
67:     msgr.postTo(key);
68:}
```

【００８８】

[SwapIn配信機構]

```
70:javax.jms.Message msg = // JMS providerから受けたメッセージをセット
71:SwapInCollection collection = // Finderメソッドで検索を行う。
72:Iterator it = collection.iterator();
73:while(true) {
74:Vector list = new Vector();
```

```
75:for(int i=0;i<20;i++) { // 20要素ずつに処理を分割して実施
76:         if (!it.hasNext()) break; //要素がなければ終了
77:     EJBObject obj = (EJBObject)it.next(); // EJBObjectを取出す
78:         AgentKey key = // objからエージェント・キーを取得する。
79:     list.addElement(key);
80:     }
81:     // このときには、上記の20エージェントはエージェント・キャッシュに
82:// 保持されている。
83:     if (list.isEmpty()) break; //もしlistが空なら終了
84:MessageListener listener = new MessageListener(list.size());
85:Messenger msgr = new Messenger(msg, listener);
86:for(int i=0;i<list.size();i++) { // listの処理を行う
87:         AgentKey key = (AgentKey)list.elementAt(i);
88:     msgr.postTo(key);// メッセージをエージェント・キューに入れる。
89:     }
90:     synchronized(listener) {
91:         // postToを行ったすべてのエージェントの処理が終了するまで待つ
。
92:         if (listener.count > 0) listener.wait();
93:     }
94:}
```

【００８９】

実施例１では、KeyOnlyCollection配信機構とSwapIn配信機構とを管理者が手動で適宜、切り替えることができるようにする。管理者が手動で切り替えることに代えて、自動切替方式を採用することもできる。自動切替機能を装備する「配信方式決定機構」について説明する。この機構は、下記に示す８つの情報（それらの一部の場合も含める）を利用して、受信したメッセージを配信するための機構として、KeyOnly配信機構を利用するかSwapIn配信機構を利用するかを決定する。どちらの機構を選択するかは、下記３から８の予測値をもとに両方の方式そ

れぞれに対してのコストを計算し、その値から決定する。

【００９０】

ここで下記８つの情報をどのような方法で取得するか、これらの情報からどのようにコストを計算するかは任意である。

（１）メッセージの属性情報（例えばメッセージのタイプを表す属性、「株価通知」など）

（２）配信すべきエージェントのタイプ

（３）メッセージを配信すべきエージェントの数の予測値（以降「N」で参照）

（４）メッセージを処理すべきエージェントを呼び出す場合のキャッシュ・ヒット率の予測値（以降「R」で参照）

（５）KeyOnly配信機構において、メッセージを受信してから、配信すべきエージェントのキー情報のリストを取得し、取得したキー情報に対応するすべてのエージェントのメッセージ・キューにメッセージを入れるのに要する時間の予測値（以降「T_k」で参照）。エージェントがメッセージを処理する時間は含まない

（６）SwapIn配信機構において、メッセージを受信してから、配信すべきエージェントのキー情報のリストを取得し、取得したキー情報に対応するすべてのエージェントのメッセージ・キューにメッセージを入れるのに要する時間の予測値（以降「T_s」で参照）。エージェントがメッセージを処理する時間は含まない

（７）キャッシュ内のエージェントに対して、メッセージの処理を実行することを決定してからそのエージェントが処理を完了するまでの処理時間の予測値（以降「t_{in}」で参照）

（８）キャッシュに読込まれていないエージェントに対して、メッセージの処理を実行することを決定してから、そのエージェントのデータをDBから読込み、そのエージェントが処理を完了するまでの処理時間の予測値（以降「t_{out}」で参照）

【００９１】

エージェント・サーバにおけるメッセージ配信機構は、メッセージを受信すると、はじめに、そのメッセージを配信すべきエージェントのタイプを決定する。その次に、配信すべきエージェントをエージェントのサブスクライブ情報を参照

して決定する。このステップで、メッセージ配信機構は、エージェントのキー情報のリストを取得するが、このときに、エージェントのキー情報のリストを取得する直前に、メッセージ配信機構は、配信方式決定機構を呼び出す。呼び出された配信方式決定機構は、受信したメッセージの属性情報と配信すべきエージェントのタイプ情報を参照して、上記（３）から（８）の情報を取得し、KeyOnly配信方式かSwapIn配信方式両方のコスト計算を行い、得られたコスト値からどちらかの方式を選択する。

【００９２】

実装例における上記８つの情報の取得方法、並びにKeyOnly配信機構及びSwapIn配信機構のコスト計算を説明する。はじめに、上記（１）から（８）に示した情報の取得方法の例を示す。

（ａ１）メッセージの属性情報：あらかじめシステム設計者や管理者が、配信方式決定に際して、メッセージのどの属性方法を参照するかを設定しておく

（ａ２）配信すべきエージェントのタイプ：メッセージを配信するときに、アプリケーションプログラムが指定する。

（ａ３）前回の値を記録しておき、その値を予測値として利用

（ａ４）前回の値を記録しておき、その値を予測値として利用

（ａ５）T_kシステム運用開始前に計測した値からNを変数とする一次方程式を求めておき、Nの予測値をその一次方程式に代入して得る。

（ａ６）T_s ：システム運用開始前に計測した値からNを変数とする一次方程式を求めておき、Nの予測値をその一次方程式に代入して得る。

（ａ７）t_{in} ：システム運用開始前に計測した値を予測値として利用

（ａ８）t_{out} ：システム運用開始前に計測した値を予測値として利用

【００９３】

ここで、（ａ３）から（ａ８）の値は、（ａ１）で設定されたメッセージの属性情報と（ａ２）で使用されるエージェントのタイプ情報の組ごとに保管される。

また、N、Rの値は前回の値を利用するため、最初の一回目は予測値が存在しない。このような場合、KeyOnly配信機構かSwapIn配信機構のどちらかをコスト計算

せずに決定する。これはどちらでも構わない。

【００９４】

次に、KeyOnly配信機構、ならびに、SwapIn配信機構のコストの計算式の例を示す。

KeyOnly配信機構のコスト（以降「T_{Key}」）

$T_{Key} = T_k + N \times R \times t_{in} + N \times (1-R) \times t_{out}$

ただし、$T_k = a_k \times N + b_k$

a_k、b_kはシステム運用開始前に計測した値から取得

【００９５】

SwapIn配信機構のコスト（以降「T_{Swap}」）

$T_{Swap} = T_s + N \times t_{in}$

ただし、$T_s = a_s \times N + b_s$

a_s、b_sはシステム運用開始前に計測した値から取得

【００９６】

配信方式決定機構は、計算されたT_{Key}、T_{Swap}の値の小さい方の配信機構を選択する。

ここで、長期間システムを運用していると、a_k、b_k、a_s、b_s、はシステム運用開始前に計測した値と大きく異なってくる場合がある。したがって、配信方式決定機構は、実行時にT_k、T_sに対応した実測値を計測しておき、必要であればa_k、b_k、a_s、b_sの値を補正する。ただし、計測値には多少の誤差はつき物であるため、逐次補正するのではなく、予測値と計測値が常に大きくずれている場合（例えば５０％以上など、この閾値は管理者が設定する）に補正を行う。また、計測値としては、他のメッセージ処理が並行して行われていない場合の値だけを使う。さもないと、計測して得られた値に他のメッセージの処理に要したCPU時間が含まれてしまうためである。メッセージ配信機構は、現在どのメッセージの配信処理を行っているかをすべて把握しているので、並行して複数のメッセージ処理が行われているかどうかを容易に判断できる。補正の行い方は、色々考えられるが、簡単な方法として、b_k、b_sの値は変化しないとして、a_k、a_sの値を再計算する方法がある。

【００９７】

（実施例２）

実施例２では、配信処理全体に要する時間の短縮を図るために、永続記憶装置からキャッシュ・メモリへのエージェントの読込み回数を減らしつつ、複数の配信起因情報の受付に対して、各配信起因情報に係る配信用メッセージが所定の配信優先度により配信されるようにする。実施例２の構成を具体的に説明する前に、実施例２の意義を理解するための背景について説明する。

【００９８】

エージェント・サーバが対象としているアプリケーションでは、処理の優先度を必要とするものがある。メッセージ処理を行うシステムでは、メッセージに優先度を付与し、そのメッセージを蓄えるキューから処理すべきメッセージを取得する時に、高い優先度のメッセージから先に取得する方法が一般的である。エージェント・サーバでは、先述したように各エージェントにメッセージ・キューを割当てる。或るエージェントのメッセージ・キューの中でメッセージを優先度順に格納すると、そのエージェント単体では高い優先度のメッセージが先に処理されるが、エージェント・サーバ全体をみると、低い優先度のメッセージを持つエージェントにスレッドが先に割当てられる場合がある。エージェント・サーバの性能を向上させるために、エージェント・キャッシュにあるエージェントから先にスレッドを割当てようとする場合において、高い優先度のメッセージをメッセージ・キューに持つエージェントがエージェント・キャッシュになく、かつそれよりも低い優先度のメッセージをメッセージ・キューに持つエージェントがエージェント・キャッシュにあるとき、後者のエージェントから先にスレッドを割当ててしまう。

【００９９】

さらに、別の優先度に関わる問題として、エージェント・サーバならではの問題がある。ここで、株価通知サービスを行うアプリケーションで、ゴールド会員と一般会員をサポートするシステムを考えてみる。株価が更新されると、エージェント・サーバでは、メッセージのFanOutを行う。これは１個のメッセージを複数のエージェントに配信することである。ここでゴールド会員に優先的に株価通

知を行いたいと考えることは珍しいことではない。しかし、既存のエージェント・サーバでは、このようなことを実現できない。既存のエージェント・サーバでは、エージェント・キャッシュの状態だけをみてエージェントのスレッド割当ての順番を決定してしまう。したがって、たまたま一般会員がエージェント・キャッシュ上にある場合、一般会員の処理が先に行われてしまい、先に通知されることになる。この問題は、単にメッセージに優先度が付与されるだけでは不十分である。なぜならば、FanOutの場合は１個のメッセージが多くのエージェントに配信されるためである。したがって、エージェントそのものにも優先度を付与し、それを考慮したスケジューリングが必要である。

【０１００】

これに対する実施例２の解決策は次の通りである。

エージェントにどのように優先度を付与するかについては任意であるが、実施例２では、エージェントに付与する優先度をエージェントのタイプに対して付与（以降「エージェント・タイプ優先度」呼ぶ。）し、そのタイプに属するエージェントはすべてその優先度が与えられる方法とする。メッセージの優先度とエージェントの優先度から、どのような順番でエージェントにスレッドを割当てるか、つまり、エージェントの実行時の優先度のつけ方は、自由である。そこで、実施例２では、エージェントの実行時の優先度（以降「エージェント実行優先度」と呼ぶ。）は、メッセージ・キュー内にあるメッセージで最大の優先度値とエージェント・タイプ優先度の単純な加算値とする。メッセージの優先度は２，１，０（２が最高）とし、エージェント・タイプ優先度は１，０（１が最高）とする。したがって、エージェント実行優先度は、３，２，１，０の４段階となる。エージェント実行優先度は、到着するメッセージの優先度に応じて動的に変化する値である。

【０１０１】

実施例２のメッセージ優先度（該メッセージ優先度は配信用メッセージの内容自体に係る優先度に相当する。）及びエージェント・タイプ優先度（該エージェント･タイプ優先度は配信元における配信先の格付けに相当する。）とは別の例として、メッセージの優先度は５．４，０（５が最高）とし、エージェント・タ

イプ優先度は１０，０（１０が最高）としてもよく、メッセージ優先度間、エージェント・タイプ優先度間、及びメッセージ優先度とエージェント・タイプ優先度との間の格差は任意に設定可能である。

【０１０２】

実施例２では、或るエージェントにいったんスレッドが割り振られると、そのエージェントのメッセージ・キューにあるメッセージを連続して処理する。ただし、ある一定個数以上連続してメッセージを処理しても、まだ残りのメッセージがある場合は、他のエージェントにスレッドを割当てる。この方法では、メッセージの優先度を厳密に反映させたスケジューリングにならないが、或るエージェントを連続して処理することで、エージェントの切り替えにかかるコストを削減できるため、システム全体の性能を向上させることができる。

【０１０３】

［データ構造］

これを行うには、個々のエージェントの管理情報を保持しなければならない。ここでは、それを管理するオブジェクトとしてControlBlock（制御ブロック）を定義する。エージェント・キューもControlBlockが管理する。エージェントのキー情報を指定することで、そのエージェントのControlBlockを取得することができる。さらに、ControlBlockは、双方向リンク構造で連結可能とする。これは、エージェント・キャッシュ内でFirstInFirstOut（FIFO:フアースト・イン・ファースト・アウト）に基づき連結するためである。この構造によりエージェントのキーを指定すると、それに該当するControlBlockを特定でき、そのメッセージ・キューにメッセージを入れることができる。ControlBlockにはエージェント・タイプ名がある。さらにエージェント・タイプとエージェント・タイプ優先度を管理するデータがある。

【０１０４】

図２１は制御ブロックのデータ構造を示している。第１の対象テーブル２３０は、エージェント・キー(AgentKey)と制御ブロック(ControlBlock)との１：１の対応関係が記録されている。エージェント・キーは、エージェントを識別するものであり、各配信先に対して１：１に対応している。図２１の第１の対象テーブ

ル２３０では、各エージェント・キーを人名（マイク(Mike)、トム(Tom)、エリック(Eric)）により表している。第２の対象テーブル２３１は、エージェント・タイプ(AgentType)と優先度(Priority)との対応関係が記録される。この例では、エージェント・タイプには、ゴールド(Gold)と一般(Normal)との２個があり、ゴールド及び一般にそれぞれ優先度(Priority)の１，０が設定されている。前述したように、ゴールド会員及び一般会員の優先度をそれぞれ例えば５，０と設定して、格差を広げることもできる。各制御ブロック２３２は、それが対応付けられているメッセージ・キューについてのデータ(MessageQueue queue)、それが連結される前側の制御ブロック２３２についてのデータ(ControlBlock prev)、それが連結される後ろ側の制御ブロック２３２についてのデータ(ControlBlock next)、それが今ある状態を表すデータ(byte state。後述のIN_FILLEDやIN_EMPTY等)、及びストリング・タイプのデータ(String type)を含む。

【０１０５】

ControlBlockにはエージェントの状態を表す変数がある。該変数の値には、RUNNING（ランニング）、IN_FILLED（イン・フィルド）、IN_EMPTY（イン・エンプティ）、OUT_FILLED（アウト・フィルド）、及びOUT_EMPTY（アウト・エンプティ）がある。図２２は制御ブロックの各変数値の意味を示している表である。また、図２３は制御ブロックの状態遷移図である。

【０１０６】

ControlBlockはエージェント実行優先度ごとにグループ化される。さらに、各グループ内では、エージェントの状態に基づいてグループ化される。このグループ化は、先に述べたリンク構造を用いて、同じグループのControlBlockをFIFOに基づいて連結することで実現される。ただし、RUNNING状態のControlBlockはこれらのグループとは別のリンクとして連結される。図２４はエージェントをその実行優先度ごとのグループ化した状態を示している。

【０１０７】

或るControlBlockのメッセージ・キューにメッセージを入れるときは次の手順で行う。

・ControlBlockにメッセージの優先度が高い順に並ぶように入れる。同じ優先度

のメッセージがある場合は、そのメッセージよりも後ろに入れる。

・挿入したメッセージがメッセージ・キューの先頭でない場合は終了。

・図２３の状態遷移図に基づいてエージェントの状態を更新する。

・エージェントのエージェント・タイプ優先度とメッセージ優先度を加算してエージェント実行優先度を求め、その値とエージェントの状態に応じたリストの最後尾にControlBlockを連結する。

・待ち状態のスレッドがある場合は、そのスレッドを再始動させる。

【０１０８】

スレッド割当てのスケジュールは次の通りである。図２４に示すデータ構造を用いて、スレッドを割当てるエージェントのControlBlockを決定する処理getNextControlBlock関数を示す。なお、関数とはＣ言語の用語であるが、処理はＪａｖａ（登録商標）等のＯＯＰ言語を使用して記述することもできる。

【０１０９】

ControlBlock getNextControlBlock() {

ステップ１：処理対象グループにエージェント実行優先度３のグループをセットする。

ステップ２：IN_FILLEDのリストが空でなければ、その先頭のControlBlockを取得して戻る。空の場合はステップ３へ。

ステップ３：OUT_FILLEDのリストが空でなければ、その先頭のControlBlockを取得して戻る。空の場合はステップ４へ。

ステップ４：処理対象グループがエージェント実行優先度０のグループの場合はNULLで戻る。そうでない場合は、処理対象グループに１個低いエージェント実行優先度をセットしてステップ２へ。

}

【０１１０】

スレッドは、いったん割当てられたエージェントのメッセージ・キューから連続してメッセージを取出し、同じエージェントの処理を行う。この連続処理はメッセージ・キューが空になるか、または、連続して処理するメッセージの個数がある制限値を超える場合、そのエージェントの処理を完了とみなし、他のエージ

ェントの処理を行う。

【０１１１】

エージェントのControlBlockの完了処理と次に処理すべきエージェントのControlBlockの取得処理は下記の手順で行われる。

ステップ１：処理が終了したエージェントのControlBlockの状態を図２３の状態遷移図に基づいて更新する。

ステップ２：メッセージ・キューの先頭のメッセージの優先度（空の場合は０とする）とそのエージェントのエージェント・タイプ優先度を加算してエージェント実行優先度を求め、その値とエージェントの状態を基に、そのControlBlockを対応するリストの最後尾に連結する。

ステップ３：getNextControlBlock関数を呼び、次のControlBlockを取得する。もし、戻り値がNULLの場合は、スレッドを待ち状態にする。

ステップ４：取得したControlBlockの状態をRUNNINGにし、対応するリストの最後尾に連結する。

ステップ５：もし、該当するエージェントがエージェント・キャッシュにない場合は、該当するエージェントのControlBlockを引数としてloadAgent関数を呼び、読込まれたエージェントをエージェント・キャッシュに置く。

ステップ６：メッセージ・キューの先頭のメッセージを取得して、エージェントのメッセージ・ハンドラを呼ぶ。

【０１１２】

また、メッセージ・キューへのメッセージの挿入処理のために待ち状態となっているスレッドが再始動された場合は、上記処理のステップ１，２を飛ばし、ステップ３，４の処理を行う。

【０１１３】

エージェント・キャッシュからのエージェントの破棄について説明する。或るエージェントをエージェント・キャッシュに読込むとき、それと同時にエージェント・キャッシュ内の或るエージェントを破棄する必要がある。破棄すべきエージェントを決定する関数getEvictedAgentの手順を以下に示す。

【０１１４】

ControlBlock getEvictedAgent() {

ステップ１：エージェント実行優先度０のグループから開始

ステップ２：IN_EMPTYのリストが空でなければ、その先頭のControlBlockの状態を図２３の状態遷移図に基づいて更新し、そのControlBlockを返す。

ステップ３：エージェント実行優先度が３の場合はステップ４へ。そうでない場合は、エージェント優先度を１個高くしてステップ２へ。

ステップ４：エージェント実行優先度０のグループから開始

ステップ５：IN_FILLEDのリストが空でなければ、その最後尾のControlBlockの状態を図２３の状態遷移図に基づいて更新し、そのControlBlockを返す。

ステップ６：エージェント優先度が３の場合はステップ７へ。そうでない場合は、エージェント実行優先度を１個高くしてステップ５へ。

ステップ７：NULLを返す。

}

【０１１５】

エージェントをエージェント・キャッシュに読込む処理であるloadAgent関数の処理手順を以下に示す。

void loadAgent(Object pkey) {

ステップ１　：エージェント・キャッシュに余裕がある場合は、pkeyで指定される対象エージェントを読込み、エージェント・キャッシュに登録し、図２３の状態遷移図に基づいてControlBlockの状態を更新して終了。

ステップ２　：getEvictedAgent関数を呼び、スワップ・アウトすべきエージェントのControlBlockを取得する。

ステップ３　：getEvictedAgent関数の戻り値がNULLの場合は、システム・エラーとして終了。

ステップ４　：getEvictedAgent関数の戻り値のControlBlockをその状態に対応するリストの最後尾に連結する。

ステップ５　：getEvictedAgent関数の戻り値のControlBlockに対応するエージェントをエージェント・キャッシュから破棄する。

ステップ６　：対象エージェントを読込み、エージェント・キャッシュに登録し

、図２３の状態遷移図に基づいてControlBlockの状態を更新して終了。

}

ステップ３において、１個もエージェント・キャッシュから破棄できないということは、エージェント・キャッシュ内のすべてのエージェントにスレッドが割当てられてRUNNING状態であることを示す。エージェント・サーバでは、スレッド数よりもエージェント・キャッシュのサイズの方がはるかに大きく構成される。したがって、システム・エラーとして扱っても問題はない。

【０１１６】

（実施例３）

実施例３の構成を具体的に説明する前に、実施例３の意義を理解するための背景について説明する。

【０１１７】

エージェント・サーバには、外部からメッセージが配信されてくる。例えば、クライアント・プログラムはMQSeriesやJMSのキューにメッセージを入れる。エージェント・サーバはこれらのキューからメッセージを受けて、あて先となるべきエージェントを特定し、そのメッセージ・キューに入れる。外部から送られてくるメッセージは連続的に送られてくることもある。これらのメッセージでは、順番が大変重要な意味を持つ場合がある。例えば、最新の株価を持つメッセージを想定してみる。あるときIBMの株価が＄100であり、次の瞬間に＄110に上昇したとする。ここで、これらを通知するメッセージが逆転してエージェントに到着すると、株価が＄110から＄100に下落したことになってしまう。したがって、エージェント・サーバに到着した順番通りに、各エージェントにメッセージ処理を行わせることは重要である。

【０１１８】

エージェント・サーバでは、各エージェントはあらかじめ自分が受け取りたいメッセージの条件をデータベースのテーブル（サブスクライブ・テーブル）に登録する。メッセージ配信機構は、このサブスクライブ・テーブルに対しSQLを発行してあて先エージェントのリストを取得して、メッセージを該当エージェントのメッセージ・キューに入れる。その後、適当なタイミングでエージェント・サ

ーバの実行環境がエージェントにスレッドを割当て、メッセージ処理を行わせる。

【０１１９】

ここで、エージェントにメッセージの到着順序通りにメッセージ処理を行わせる最も簡単な方法は、外部からのメッセージの受信とエージェントへのメッセージの配信を行うスレッドを１個だけにすることである。こうすれば、必ず受信した順番にメッセージを各エージェントのメッセージ・キューに入れることができる。しかし、この方法では、並行処理が行えず、結果としてメッセージ配信処理のスループットが低下する。メッセージ配信機構は、データベースにアクセスする。データベースから結果が戻るまで、エージェント・サーバのメッセージ配信機構は完全に停止してしまうのである。

【０１２０】

そこで、メッセージ配信機構に複数のスレッドを割当て、それぞれのスレッドでメッセージの受信処理とメッセージ配信処理を行わせれば、複数のメッセージのメッセージ配信処理を同時並行して行うことができる。データベースにも並行してアクセスでき、スループットの向上が期待できる。しかし、この方法では、最初に受信したメッセージの処理を行うスレッドと、その次に受信したメッセージの処理を行うスレッドのどちらが先にメッセージをエージェントのメッセージ・キューに入れるかが分からなくなる。メッセージは、そのメッセージが持つ情報とサブスクライブ・テーブルの検索結果から決定される。そのため、メッセージごとにあて先となるエージェントは異なり、或るメッセージのあて先エージェント数が１０万で、その直後のメッセージのあて先エージェント数は１００である場合もある。このような状況では、或るエージェントのメッセージ・キューに入るメッセージの順番の逆転は容易に起こりうる。

【０１２１】

実施例３では、配信起因情報に係る配信用メッセージを、配信先へ配信起因情報の受付け順に配信することを保障する。図２５は配信起因情報に係る配信用メッセージを、配信先へ配信起因情報の受付け順に配信することを保障するエージェント・サーバの構成図である。各符号が指している要素は次の通りである。

２５０：実施例３に係るエージェント・サーバ

２５１：キュー・マネージャ(QueueManager)

２５２：メッセージ受信部(MessageReceiver)

２５３：メッセージ受信部２５２が使用するスレッド

２５４：メッセンジャ

２５５：メッセージ順番記録器(MessageOrderRecorder)

２５８：メッセージ・リゾルバ

２５９：メッセージ・リゾルバ２５８が使用するスレッド

２６３：スクライブ・テーブル

２６４：エージェント・メッセージ・キュー

２６８：エージェント・ビーン

２６９：エージェント・スケジューラ

２７０：エージェント・スケジューラ２６９が使用するスレッド

【０１２２】

QueueManagerは、エージェント・サーバ外部にあるメッセージを蓄え、エージェント・サーバに配信する機構である。QueueManagerから配信されるメッセージを受信するコンポーネントがMessageReceiverである。これは１個のスレッドで動いており、メッセージを受信するとMessengerオブジェクトを生成し、そのオブジェクトに受信したメッセージをセットする。さらに、通番をセットする。その後にMessageOrderRecorderにMessengerオブジェクトを渡す。

【０１２３】

ここで、Messengerオブジェクトとは、外部からのメッセージのオブジェクトを保持すると同時に、図２６の表にある状態を保持するオブジェクトであり、エージェントのメッセージ・キューにはMessengerオブジェクトが入れられる。図２６はメッセンジャ・オブジェクトの各処理状態を説明した表である。

【０１２４】

MessageOrderRecorderはMessengerオブジェクトを順番通りに記録するコンポーネントである。MessageOrderRecorderは、その状態を"NotProcessed"にして通番の小さい順にソートして記録する。

MessageReceiverはMessengerオブジェクトをMessageOrderRecorderにセットした後、MessageResolverを呼び出し、Messengerオブジェクトの処理を依頼する。

【０１２５】

この依頼を受けたMessageResolverは、自分が管理するスレッド群の中からアイドル中のスレッドを取得し、それを起こす（wake）。このとき、もしアイドル中のスレッドがなければ（つまりすべてのスレッドが処理中である状況）、何もしない。MessageResolverのスレッドはMessageOrderRecorderから未処理のMessengerオブジェクトを取得する。このとき、MessageOrderRecorderはMessengerオブジェクトの状態が"NotProcessed"のもので、通番がもっとも小さいMessengerオブジェクトを選択し、そのMessengerオブジェクトに処理中を意味する"Distributing"状態をセットする。MessageOrderRecorderはこれらの処理を同時並行的に処理が行われないようにしなければならない。このスレッドはMessageResolverを呼び出し、取得したMessengerオブジェクトの処理を行うように依頼する。

【０１２６】

呼び出されたMessageResolverは、データベースにあるサブスクライブ・テーブルに対し検索処理を行い、メッセージを渡すべきエージェントのエージェント・キーの集合を取得する。次に、エージェント・キーに対応するエージェントのメッセージ・キューにMessengerオブジェクトを入れる。すべてのあて先エージェントのメッセージ・キューにMessengerオブジェクトの挿入が終了すると、MessageOrderRecorderを呼び出し、Messengerオブジェクトの挿入処理が完了したことを知らせる。Messengerオブジェクトをメッセージ・キューに入れるとき、そのメッセージ・キューにすでにあるMessengerオブジェクトの通番を参照して、その値が小さい順にソートされるように入れる。

【０１２７】

MessageOrderRecorderは、そのMessengerオブジェクトの通番を参照して、それより小さい番号のMessengerオブジェクトがある場合は、Messengerオブジェクトの状態を"Distributed"にする。このようなMessengerオブジェクトがない場合は、そのMessengerオブジェクトの状態を"Ready"にし、同時にMessageOrderRecorderからそのMessengerオブジェクトの記録を消去する。さらに、このMessenger

オブジェクトの通番＋１の通番を持つMessengerオブジェクトの状態を参照し、それが"Distributed"である場合は、そのMessengerオブジェクトの状態を"Ready"にし、同時にMessageOrderRecorderからそのMessengerオブジェクトの記録を消去する。同様の処理をMessengerオブジェクトの状態が"Distributing"であるものに出会うまで繰り返す。最終的に、"Distributing"状態であるMessengerオブジェクトに出会ったら、そのMessengerオブジェクトの状態を"Ready"にする。MessageOrderRecorderはこれらの処理を同時並行的に処理が行われないようにしなければならない。

【０１２８】

一方、AgentSchedulerは各エージェントのメッセージ・キューの状況をみてスレッドを割当てる。AgentSchedulerは、メッセージ・キューの先頭にあるMessengerオブジェクトの状態が"Ready"であれば、そのメッセージ・キューに対応したエージェントにスレッドを割当てる。そうでない場合は、他のエージェントを確認する。各コンポーネントの処理手順を以下に示す。

【０１２９】

[MessageReceiverの受信処理]

1: メッセージを受信する。

2: Messengerオブジェクトを生成し、メッセージをセットする。

3: Messengerオブジェクトに通番をセットする。

4: Messengerオブジェクトを引数としてMessageOrderRecorderの記録処理を呼び出す

5: MessageResolver用のスレッド群にもしアイドル中のスレッドがあれば、それを再起動させる。

【０１３０】

[MessageOrderRecorderの記録処理]

1: 受け取ったMessengerオブジェクトの状態を"NotProcessed"にして通番の小さい順にソートして記録する。

【０１３１】

[MessageOrderRecorderの取得処理]

1: 記録にあるMessengerオブジェクトで、状態が"NotProcessed"のもののうち最も通番の小さいMessengerオブジェクトを取得する。
2: そのオブジェクトが現在記録されているすべてのMessengerオブジェクトの中で通番が最小の場合、その状態を"Ready"にする。そうでない場合、その状態を"Distributing"にする。
3: そのMessengerオブジェクトを返す

【０１３２】

[MessageOrderRecorderの挿入完了処理]

1: 渡されたMessengerオブジェクトの通番より小さい通番を持つMessengerオブジェクトが記録されている場合、渡されたMessengerオブジェクトの状態を"Distributed"にして処理から抜ける。
2: 渡されたMessengerオブジェクトの状態を"Ready"にして記録から消去する。
3: 対象Messengerオブジェクトとして、渡されたMessengerオブジェクトの次に記録されているMessengerオブジェクトをセットする。
4: 対象Messengerオブジェクトの状態が"Distributing"であれば、処理を抜ける。
5: 対象Messengerオブジェクトが"Distributed"である場合、対象Messengerオブジェクトの状態を"Ready"にする。
6: 対象Messengerオブジェクトを記録から消去する。
7: 対象Messengerオブジェクトを現在の対象Messengerオブジェクトの次のものにセットして、ステップ４に戻る。

【０１３３】

[MessageResolverのメッセージ挿入処理]

1: MessageOrderRecorderの取得処理を呼び出し、処理すべきMessengerオブジェクトを取得する。もしなければ、この処理から抜ける。
2: 取得したMessengerオブジェクトのメッセージのデータを参照してサブスクライブ・テーブルに検索処理を行い、メッセージを渡すべきエージェントのエージェント・キーのリストを取得する。
3: 取得したリストのすべてのエージェント・キーに対して、その個々のエージ

エント・キーに対応するエージェントのメッセージ・キューにMessengerオブジェクトを挿入する。このとき、そのメッセージ・キューにMessengerオブジェクトがすでにある場合、その通番の小さい順にソートされるように挿入する。

4: Messengerオブジェクトを引数として、MessageOrderRecorderの挿入完了処理を呼ぶ

【０１３４】

[AgentSchedulerのスレッド割当て処理]

1: エージェント・サーバが管理するエージェントのリストから、メッセージ・キューが空でないエージェントを選択する。

2: このエージェントのメッセージ・キューの先頭のMessengerオブジェクトの状態が"Ready"であれば、先頭のMessengerオブジェクトからメッセージを取得し、そのエージェントのメッセージハンドラを呼ぶ。そうでない場合はステップ１に戻る。

3: ステップ２をそのメッセージ・キューが空になるか、"Ready"でないMessengerオブジェクトに出会うまで繰り返す

4: ステップ１に戻る。

【０１３５】

（実施例４）

パソコン関連商品を扱う電子モール・システムにおける本発明の実施例を示す。この実施例では、複数の商品の合計金額が利用者の設定した金額内に納まる場合に、利用者に情報を電子メールで配信しする。この実施例では、最終的には電子メールが利用者に配信されるが、その途中の過程では、エージェントは電子モールの配信を行わない処理を実行する。

【０１３６】

ある日、田中さんは、６００ＭＨｚ以上のＣＰＵを搭載したパソコンを１台、１２８ＭＢの拡張メモリを２個、カラー・プリンタを１台、合計１０万円以内で購入しようとしたが、８万円のパソコンと１個につき１万円の１２８ＭＢのメモリ、１万８千円のカラー・プリンタしか見つからず、予算額を超えるため、合計１０万円以内になるような製品の組み合わせが見つかったら電子メールで通知し

てくれるようにシステムに入力した。２日後、この電子モールでは７万円の６００ＭＨｚのＣＰＵを持つベータ社のパソコンＢ－６００が登録され、その翌日、１万円のエジソン社のカラー・プリンタが登録された。するとベータ社パソコンＢ－６００＋エジソン社カラー・プリンタ＋１万円の１２８ＭＢメモリ×２で合計１０万円になるので、田中さんに「ベータ社のパソコンＢ－６００と１２８ＭＢメモリ２つとエジソン社のカラー・プリンタで合計１０万円」という文面の電子メールが送られた。

【０１３７】

このシステムの内部では次のような処理がなされる。田中さんは電子モール・システム内にいる田中さん用のエージェントに対して、「６００ＭＨｚ以上のパソコン＋１２８ＭＢメモリ２枚＋カラー・プリンタで合計１０万円以内になるような製品の組み合わせが見つかったら電子メールで通知してくれ」という主旨の条件をＷｅｂブラウザから入力する。するとこのエージェントは、商品情報ＤＢからパソコン、メモリ、プリンタ情報を取得する。この場合では、田中さんのエージェントは「８万円のパソコン」、１つ「１万円の１２８ＭＢのメモリ」、「１万８千円のカラー・プリンタ」の情報を取得し、その情報をエージェントがデータとして保持する。これと同時に、サブスクライブ情報として、「パソコン」、「メモリ」、「カラー・プリンタ」の情報を登録する。２日後、商品情報データベースに７万円のベータ社のパソコン情報が入力される。これと同時に、パソコン情報が更新されたことを通知するイベントがシステムに送られる。すると、このイベントはエージェント・サーバのメッセージ機構により、「７万円の６００ＭＨｚのＣＰＵを持つベータ社のパソコンＢ－６００」の情報を持つ新着商品メッセージというメッセージに変換され、関連するエージェントすべてのメッセージキューに挿入される。田中さんエージェントはこのメッセージに関連するため、田中さんエージェントのメッセージキューにもこのメッセージが挿入される。その後、田中さんエージェントのメッセージハンドラが呼び出され、その引数としてこのメッセージが渡される。これを受けた田中さんエージェントは、そのエージェントが保持している「１万円の１２８ＭＢのメモリ」、「１万８千円のカラー・プリンタ」との合計金額を計算する。しかし、合計金額は１０万８千円で

あり、まだ１０万円を超えているので、電子メールの送信は行わない。その代わり、田中さんエージェントが保持している商品データのリストに「７万円の６００ＭＨｚのＣＰＵを持つベータ社のパソコンＢ－６００」を追加する。３日後、「１万円のエジソン社のカラー・プリンタ」情報のイベントが同様の方法でエージェントに送られる。この時、田中さんのエージェントのメッセージハンドラが呼び出され、引数として「１万円のエジソン社のカラー・プリンタ」情報のメッセージが渡される。ここで、「ベータ社パソコンＢ－６００＋エジソン社カラー・プリンタ＋１万円の１２８ＭＢメモリ×２」で合計額が１０万円となるので田中さんエージェントは田中さんに「ベータ社のパソコンＢ－６００と１２８ＭＢメモリ２つとエジソン社のカラー・プリンタで合計１０万円」という文面の通知メールを送信する。

【０１３８】

この実施例でのポイントは、田中さんエージェントが「７万円の６００ＭＨｚのＣＰＵを持つベータ社のパソコンＢ－６００」のメッセージをメッセージハンドラで処理するときに、電子メールの送信は行わないということである。田中さんエージェントが実行する処理は次のものである。

１．：すでに保持しているメモリとプリンタとの合計金額を計算

２．：その合計金額と「１０万円」という値と比較

３．：その結果電子メールを送信しないことを決定

４．：「７万円の６００ＭＨｚのＣＰＵを持つベータ社のパソコンＢ－６００」をデータに追加

【０１３９】

（実施例５）

実施例５では、処理要求元が、人や組織等の会員の概念に属しないものについて説明する。企業Ａが複数の企業からパソコン部品の受注を受ける企業間ワークフローシステムを考える。このシステムは、発注元企業（下記の例での企業Ｂと企業Ｃ）の発注システムと企業Ａの受注システムが連携されたシステムである。

【０１４０】

９月２０日に企業Ｂが発注番号００１２で「モデル番号１１２４のＨＤＤ（ハ

ード･ディスク・ドライブ装置）１００台を、納入希望期限を１０月１日、納入最終期限を１０月３日として発注」を依頼する処理を開始する。また、同日に企業Ｃが発注番号００１３で「モデル番号１１２４のＨＤＤ２００台を、納入希望期限を１０月５日、納入最終期限を１０月１０日として発注」を依頼する処理を開始する。

【０１４１】

すると、企業Ａのワークフローシステムでは、発注番号００１２ワークフロー処理、発注番号００１３ワークフロー処理が開始される。これと同時に、企業Ａのエージェント・サーバにおいて、発注番号００１２ワークフロー処理と発注番号００１３ワークフローを監視するための監視エージェントである発注番号００１２監視エージェントと発注番号００１３監視エージェントが生成される。このエージェント・サーバは企業Ａのワークフローシステムと連携されたものである。

ワークフローを監視する監視エージェントは、下記の監視を行う。

１．：発注ワークフロー処理がどこかの処理で遅延していないか

２．：何らかの理由でこのワークフロー処理が失敗したか

３．：何らかの理由でこのワークフロー処理を無効化しなければならないか

４．：依頼元である企業が発注処理のキャンセル処理を開始したか

【０１４２】

ここで、９月２６日に「当初モデル番号１１２４のＨＤＤは企業Ａにその下請け企業又は系列企業等の関連企業から９月２９日までに５００台納入される予定であったが、モデル番号１１２４のＨＤＤの生産が遅れ、企業Ａに１０月６日に５００台納入」となったとする。この場合、この情報をあらわすイベントがエージェント・サーバに送られる。エージェント・サーバでは、このイベントを「当初モデル番号１１２４のＨＤＤは企業Ａに９月２９日までに５００台納入される予定であったが、モデル番号１１２４のＨＤＤの生産が遅れ、企業Ａに１０月６日に５００台納入」を示すメッセージに変換し、モデル番号１１２４ＨＤＤの発注ワークフローを監視している全ての監視エージェントに渡される。発注番号００１２監視エージェントと発注番号００１３監視エージェントにもこのメッセー

ジが渡される。ここで、発注番号００１２監視エージェントのメッセージハンドラでは、発注番号００１２ワークフロー処理を無効化し、企業Ｂのシステムに「発注番号００１２をキャンセルした」という通知メッセージを送る。発注番号００１３監視エージェントでは、発注番号００１３ワークフローはそのまま継続させるが、企業Ｃのシステムに「発注番号００１３の納入期限が１０月６日以降に変更」という通知メッセージを送る。

【０１４３】

実施例５では、エージェントは会員に対して生成されるものではなく、「ワークフロー処理」に対して生成される。また、エージェントのメッセージの通知先は「人」ではなく、コンピュータシステムである。この実施例で挙げた企業Ａにおけるワークフロー処理システムでは、ワークフロー処理手続きは大量に発生する。一方、個々のワークフロー処理手続きでは、納入期限や扱う商品などがすべて異なる。そのため、「当初モデル番号１１２４のＨＤＤは企業Ａに９月２９日までに５００台納入される予定であったが、モデル番号１１２４のＨＤＤの生産が遅れ、企業Ａに１０月６日に５００台納入」のようなイベントが発生した場合、そのイベントに対して、どのような処理を行うべきかはワークフロー処理ごとに異なる。そのため、個々のワークフロー処理ごとに監視エージェントを生成する方法が考えられる。この場合、システムは大量の監視エージェントを扱う必要が生じる。また、先の例で示したように大量エージェントのメッセージ処理も必要になる。

【０１４４】

まとめとして本発明の構成に関して以下の事項を開示する。

（１）：エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置から

プログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能でとなる複数個のエージェント、

前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込み手段、

メッセージが挿入されているメッセージ・キューに係るエージェントにその作動を指示するエージェント用指示手段、及び

前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込み手段にその処理の繰返しを指示する繰返し指示手段、

を有していることを特徴とするメッセージ処理装置。

（２）：各エージェントは、該エージェントに対応付けられているメッセージ・キューにおけるメッセージに対して、該エージェントに対応付けられた処理要求元への通知を実施するものであることを特徴とする（１）記載のメッセージ処理装置。

【０１４５】

（３）：エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・

メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能でとなる複数個のエージェント、

第１のメッセージ・キュー用処理機構、

第２のメッセージ・キュー用処理機構、

第１及び第２のメッセージ・キュー用処理機構のどちらかを選択する選択手段、及び

メッセージが挿入されているメッセージ・キューに係るエージェントについて該エージェントがキャッシュ・メモリに有れば該エージェントにその作動を直ちに指示しまた該エージェントがキャッシュ・メモリに無ければ該エージェントを前記永続記憶装置から前記キャッシュ・メモリへ読込んでから該エージェントにその作動を指示するエージェント用指示手段、

を有し、

第１のメッセージ・キュー用処理機構は、

前記リスト情報に含まれる全部の処理要求元に係るメッセージ・キューに前記メッセージを挿入する挿入手段、

を有し、

第２のメッセージ・キュー用処理機構は、

前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込み手段、及び

前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込み手段にその処理の繰返しを指示する繰返し指示手段、

を有している、

ことを特徴とするメッセージ処理装置。

【０１４６】

（４）：前記選択手段の選択はオペレータの指示に基づくことを特徴とする（３）記載のメッセージ処理装置。

（５）：前記選択手段の選択は、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元の予測数、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元に係るエージェントについてのキャッシュ・メモリにおける予測ヒット率、前記選択手段が第１のメッセージ・キュー用処理機構を選択している場合に前記受付手段が情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェントのメッセージ・キューに挿入するまでの予測作業時間、前記選択手段が前記第２のメッセージ・キュー用処理機構を選択している場合に前記受付手段が情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェントのメッセージ・キューに挿入するまでの予測作業時間、キャッシュ・メモリ内のエージェントについてその使用を決定してから該決定したエージェントが処理完了するまでの予測時間、及び／又はキャッシュ・メモリ外のエージェントについてその使用を決定してから該決定したエージェントが処理完了するまでの予測時間に基づくことを特徴とする（３）記載のメッセージ処理装置。

【０１４７】

（６）：エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元を前記処理要求元検索情報に基づいて決定する処理要求元決定手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているとき作動可能となる複数個のエージェント、

各メッセージについての処理優先度を単一の価値基準に基づいてサブ処理優先

度として決定する少なくとも１個のサブ処理優先度決定手段、

前記サブ処理優先度決定手段の総個数が２以上であるときは各サブ処理優先度決定手段が各メッセージについて個々に決定したサブ処理優先度に基づいて各メッセージについての複合処理優先度を決定し、また、前記サブ処理優先度決定手段の総個数が１であるときは該唯一のサブ処理優先度決定手段が各メッセージについて決定したサブ処理優先度を複合処理優先度として決定する複合処理優先度決定手段、

各メッセージ・キューが保持しているメッセージの中で最高複合処理優先度のメッセージを最優先メッセージとし該最優先メッセージの複合処理優先度が同一であるメッセージ・キューに係るエージェント同士では、キャッシュ・メモリに存在する方のエージェントを、存在しない方のエージェントより優先してその作動を指示するエージェント用指示手段、

を有していることを特徴とするメッセージ処理装置。

【０１４８】

（７）：前記価値基準には、メッセージの内容に係るもの及びメッセージが適用される処理要求元に係るものがあること特徴とする（６）記載のメッセージ処理装置。

（８）：メッセージの内容に係る所定の価値基準には、メッセージの処理の緊急性に係る価値基準が含まれることを特徴とする（７）記載のメッセージ処理装置。

（９）：メッセージが適用される処理要求元に係る価値基準には、処理要求元の格付けに係る価値基準が含まれることを特徴とする（７）記載のメッセージ処理装置。

（１０）：前記エージェント用指示手段により処理を一旦、開始したエージェントは、該エージェントに係るメッセージ・キューが保持するメッセージが複数個あるとき、それら全部のメッセージについて、又は複合処理優先度が上位から所定番目までのメッセージについて連続処理することを特徴とする（６）～（９）のいずれかに記載のメッセージ処理装置。

（１１）：前記サブ処理優先度決定手段及び前記複合処理優先度決定手段を含む

エージェント管理手段を有し、

前記エージェント管理手段は、各エージェントが永続記憶装置に存在するか否かを検出する存在検出手段、該存在検出手段の判定結果及び各エージェントの複合処理優先度に基づいてエージェントをグループ化しグループ化情報を管理するグループ化情報管理手段、及びグループ化情報管理手段にグループ化情報の更新を指示する更新指示手段を有し、

前記エージェント用指示手段は、前記エージェント管理手段におけるグループ化情報に基づく順番でエージェントにその作動を指示する、

ことを特徴とする（６）～（１０）のいずれかに記載のメッセージ処理装置。

【０１４９】

（１２）：エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理手段、

エージェント起動原因イベントを受付ける受付手段、

前記受付手段が受付けたエージェント起動原因イベントについての受付順番情報を管理する受付順番情報管理手段、

各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能となっているエージェントであって各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能となっている複数個のエージェント、

相互に並列動作可能である複数個のスレッドであって各スレッドはエージェント起動原因イベントを起因とするメッセージが適用される処理要求元を処理要求元検索情報に基づいて検出しかつ各検出処理要求元に係るメッセージ・キューに前記メッセージを挿入する複数個のスレッド、

各スレッドにそれが処理を担当するエージェント起動原因イベントを割当てる割当て手段、

前記受付手段が受付けた各エージェント起動原因イベントについてのスレッドによる処理の進行状態情報を管理する進行状態情報管理手段、

処理進行状態情報がスレッド処理終了に係る情報になっているエージェント起動原因イベント（以下、「被判定エージェント起動原因イベント」と言う。）について、該被判定エージェント起動原因イベントより前に前記受付手段において受付けたエージェント起動原因イベントの中にまだスレッド処理の未終了になっているエージェント起動原因イベントが有るか無いかを判定する判定手段、及び

前記判定手段が「有る」と判定した被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェントによる処理を抑制するエージェント制御手段、

を有していることを特徴とするメッセージ処理装置。

【０１５０】

（１３）：前記判定手段が「無い」と判定した被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェントによる処理を許容する前記エージェント制御手段、

を有していることを特徴とする（１２）記載のメッセージ処理装置。

（１４）：「無い」と判定したエージェント起動原因イベントに対して受付順番がその次のエージェント起動原因イベントが、「有る」との判定済みのものであるときは、判定結果を「有る」から「無い」へ変更する前記判定手段、

を有していることを特徴とする（１３）記載のメッセージ処理装置。

（１５）：前記判定手段による判定結果が「無い」となっている複数個のエージェント起動原因イベントを生成起因とするメッセージが受付順番方向へ連続するメッセージ･キューについての処理を実行する場合は、それら連続する複数個のメッセージについての処理を連続的に行う前記エージェント、

を有していることを特徴とする（１４）記載のメッセージ処理装置。

【０１５１】

（１６）：エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情

報作成ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能であり各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能であるエージェント設定ステップ、

前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込みステップ、

メッセージが挿入されているメッセージ・キューに係るエージェントにその作動を指示するエージェント用指示ステップ、及び

前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込みステップの繰返しを指示する繰返し指示ステップ、

を有していることを特徴とするメッセージ処理方法。

（１７）：各エージェントは、該エージェントに対応付けられているメッセージ・キューにおけるメッセージに対して、該エージェントに対応付けられた処理要求元への通知を実施するものであることを特徴とする（１６）記載のメッセージ処理装置。

【０１５２】

（１８）：エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元のリスト情報を前記処理要求元検索情報に基づいて作成するリスト情報作成ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは各処理要求元に対応付けられており永続記憶装置に記憶され永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能であり各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能であるエージェント設定ステップ、

第１のメッセージ・キュー用処理ステップ、

第２のメッセージ・キュー用処理ステップ、

第１及び第２のメッセージ・キュー用処理ステップのどちらかを選択する選択ステップ、及び

メッセージが挿入されているメッセージ・キューに係るエージェントについて該エージェントがキャッシュ・メモリに有れば該エージェントにその作動を直ちに指示しまた該エージェントがキャッシュ・メモリに無ければ該エージェントを前記永続記憶装置から前記キャッシュ・メモリへ読込んでから該エージェントにその作動を指示するエージェント用指示ステップ、

を有し、

第１のメッセージ・キュー用処理ステップは、

前記リスト情報に含まれる全部の処理要求元に係るメッセージ・キューに前記メッセージを挿入する挿入ステップ、

を有し、

第２のメッセージ・キュー用処理ステップは、

前記リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し該挿入・読込み対象処理要求元に係るメッセージ・キューに、前記メッセージを挿入するとともに、前記挿入・読込み対象処理要求元に係るエージェントを永続記憶装置からキャッシュ・メモリへ読込む挿入・読込みステップ、及び

前記リスト情報に含まれる処理要求元の中に未選択のものが残っている場合には、作動中の全部のエージェントの処理終了を待って前記挿入・読込みステップの繰返しを指示する繰返し指示ステップ、

を有している、

ことを特徴とするメッセージ処理方法。

【０１５３】

（１９）：前記選択ステップにおける選択はオペレータの指示に基づくことを特徴とする（１８）記載のメッセージ処理方法。

（２０）：前記選択ステップにおける選択は、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元の予測数、今回のエージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元に係るエージェントについてのキャッシュ・メモリにおける予測ヒット率、第１のメッセージ・キュー用処理ステップにおいて処理を割当てられた場合に前記受付ステップにおいて情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェントのメッセージ・キューに挿入するまでの予測作業時間、第２のメッセージ・キュー用処理ステップにおいて処理を割当てられた場合に前記受付ステップにおいて情報を受け取ってからメッセージが適用される処理要求元のリスト情報を取得しかつメッセージを全部のエージェントのメッセージ・キューに挿入するまでの予測作業時間、キャッシュ・メモリ内のエージェントについてその使用を決定してから該決定したエージェントが処理完了するまでの予測時間、及び／又はキャッシュ・メモリ外のエージェントについてその使用を決定してから該決定したエージェントが処理完了するまでの予測時間に基づくことを特徴とする（１８）記載のメッセージ処理方法。

【０１５４】

（２１）：エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記エージェント起動原因イベントを生成起因とするメッセージが適用される処理要求元を前記処理要求元検索情報に基づいて決定する処理要求元決定ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージ

ェントは各処理要求元に対応付けられており永続記憶装置に記憶され各エージェントは永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能でありかつ各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能となっているように設定するエージェント設定ステップ、

各メッセージについての処理優先度を単一の価値基準に基づいてサブ処理優先度として決定する少なくとも１個のサブ処理優先度決定ステップ、

前記サブ処理優先度決定ステップの総個数が２以上であるときは各サブ処理優先度決定ステップにおいて各メッセージについて個々に決定したサブ処理優先度に基づいて各メッセージについての複合処理優先度を決定し、また、前記サブ処理優先度決定ステップの総個数が１であるときは該唯一のサブ処理優先度決定ステップにおいて各メッセージについて決定したサブ処理優先度を複合処理優先度として決定する複合処理優先度決定ステップ、及び

各メッセージ・キューが保持しているメッセージの中で最高複合処理優先度のメッセージを最優先メッセージとし該最優先メッセージの複合処理優先度が同一であるメッセージ・キューに係るエージェント同士では、キャッシュ・メモリに存在する方のエージェントを、存在しない方のエージェントより優先してその作動を指示するエージェント用指示ステップ、

を有していることを特徴とするメッセージ処理方法。

【０１５５】

（２２）：前記価値基準には、メッセージの内容に係るもの及びメッセージが適用される処理要求元に係るものがあること特徴とする（２１）記載のメッセージ処理方法。

（２３）：メッセージの内容に係る所定の価値基準には、メッセージの処理の緊急性に係る価値基準が含まれることを特徴とする（２２）記載のメッセージ処理方法。

（２４）：メッセージが適用される処理要求元に係る価値基準には、処理要求元の格付けに係る価値基準が含まれることを特徴とする（２２）記載のメッセージ

処理方法。

（２５）：前記エージェント用指示ステップにより処理を一旦、開始したエージェントは、該エージェントに係るメッセージ・キューが保持するメッセージが複数個あるとき、それら全部のメッセージについて、又は複合処理優先度が上位から所定番目までのメッセージについて連続処理することを特徴とする（２１）記載のメッセージ処理方法。

（２６）：前記サブ処理優先度決定ステップ及び前記複合処理優先度決定ステップを含むエージェント管理ステップを有し、

前記エージェント管理ステップは、各エージェントが永続記憶装置に存在するか否かを検出する存在検出ステップ、及び該存在検出ステップの判定結果及び各エージェントの複合処理優先度に基づいてエージェントをグループ化しグループ化情報を管理するとともに該グループ化情報を適宜更新するグループ化情報管理ステップを有し、

前記エージェント用指示ステップは、前記エージェント管理ステップにおけるグループ化情報に基づく順番でエージェントにその作動を指示する、

ことを特徴とする（２１）～（２５）のいずれかに記載のメッセージ処理方法。

【０１５６】

（２７）：エージェント起動原因イベントに対して該当処理要求元を検索するための処理要求元検索情報を管理する処理要求元検索情報管理ステップ、

エージェント起動原因イベントを受付ける受付ステップ、

前記受付ステップにおいて受付けたエージェント起動原因イベントについての受付順番情報を管理する受付順番情報管理ステップ、

複数個のエージェントを設定するエージェント設定ステップであって各エージェントは、各処理要求元に対応付けられており、永続記憶装置に記憶され、永続記憶装置からプログラム実行領域としてのキャッシュ・メモリへ読込み可能及びキャッシュ・メモリから破棄可能であり、かつ各エージェントはキャッシュ・メモリに存在しているときのみ作動して該エージェントに対応のメッセージ・キュー内のメッセージについての処理を実施可能となっているように設定するエージェント設定ステップ、

複数個のスレッドを設定するスレッド設定ステップであって各スレッドは、相互に並列動作可能であり、エージェント起動原因イベントを起因とするメッセージが適用される処理要求元を処理要求元検索情報に基づいて検出し、かつ各検出処理要求元に係るメッセージ・キューに前記メッセージを挿入するスレッド設定ステップ、

各スレッドにそれが処理を担当するエージェント起動原因イベントを割当てる割当てステップ、

前記受付ステップにおいて受付けた各エージェント起動原因イベントについての各スレッドによる処理の進行状態情報を管理する進行状態情報管理ステップ、

処理進行状態情報がスレッド処理終了に係る情報になっているエージェント起動原因イベント（以下、「被判定エージェント起動原因イベント」と言う。）について、該被判定エージェント起動原因イベントより前に前記受付ステップにおいて受付けたエージェント起動原因イベントの中にまだスレッド処理の未終了になっているエージェント起動原因イベントが有るか無いかを判定する判定ステップ、及び

前記判定ステップにおいて「有る」と判定された被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェントによる処理を抑制するエージェント制御ステップ、

を有していることを特徴とするメッセージ処理方法。

【０１５７】

（２８）：前記判定ステップにおいて「無い」と判定された被判定エージェント起動原因イベントを生成起因とするメッセージについてのエージェントによる処理を許容する前記エージェント制御ステップ、

を有していることを特徴とする（２７）記載のメッセージ処理方法。

（２９）：「無い」と判定されたエージェント起動原因イベントに対して受付順番がその次のエージェント起動原因イベントが、「有る」との判定済みのものであるときは、判定結果を「有る」から「無い」へ変更する前記判定ステップ、

を有していることを特徴とする（２８）記載のメッセージ処理方法。

（３０）：前記判定ステップによる判定結果が「無い」となっている複数個のエ

ージェント起動原因イベントを生成起因とするメッセージが受付順番方向へ連続するメッセージ･キューについての処理をエージェントに実行させる場合、該エージェントにはそれら連続する複数個のメッセージについての処理を連続的に行わせるように前記エージェントを設定する前記エージェント設定ステップ、
を有していることを特徴とする（２９）記載のメッセージ処理方法。
（３１）：（１６）～（３０）のいずれかのメッセージ処理方法における各ステップをコンピュータに実行させることを特徴とするメッセージ処理プログラム。

【０１５８】

【発明の効果】

本発明によれば、受付けたエージェント起動原因イベントを起因とするメッセージが適用される処理要求元を処理要求元検索情報に基づき割り出して、該当する各処理要求元に対応付けられたエージェントについて永続記憶装置からキャッシュ・メモリへの読込みを効果的に抑制し、処理時間の短縮を実現することができる。

【図面の簡単な説明】

【図１】

メッセージ処理システムの概略図である。

【図２】

メッセージ処理装置の構成図である。

【図３】

エージェントの構成図である。

【図４】

別のメッセージ処理装置の構成図である。

【図５】

第１の配信制御機構の詳細図である。

【図６】

第２の配信制御機構の詳細図である。

【図７】

メッセージ処理装置の構成図である。

【図８】

図７のサブ配信優先度決定手段及び複合配信優先度決定手段を構成要素として含むエージェント管理手段の全体構成図である。

【図９】

メッセージ処理装置の構成図である。

【図１０】

メッセージ処理方法のフローチャートである。

【図１１】

図１０のフローチャートにおける挿入・読込みステップの詳細図である。

【図１２】

別のメッセージ処理方法のフローチャートである。

【図１３】

図１２の第１の配信制御ステップの具体的な処理を示す図である。

【図１４】

図１２の第２の配信制御ステップの具体的な処理を示す図である。

【図１５】

さらに別のメッセージ処理方法のフローチャートである。

【図１６】

図１５のサブ配信優先度決定ステップ等を含む上位ステップを備えるメッセージ処理方法部分のフローチャートである。

【図１７】

さらに別のメッセージ処理方法のメッセージ処理方法のフローチャートである。

【図１８】

図１７のエージェント制御ステップの配信処理制御の具体例を示す図である。

【図１９】

エージェント・サーバの概略構成図である。

【図２０】

エージェント・サーバによる種々の配信方法についての説明図である。

【図２１】

制御ブロックのデータ構造を示す図である。

【図２２】

制御ブロックの各変数値の意味を示している表である。

【図２３】

制御ブロックの状態遷移図である。

【図２４】

エージェントをその実行優先度ごとのグループ化した状態を示す図である。

【図２５】

配信起因情報に係る配信用メッセージを、配信先へ配信起因情報の受付け順に配信することを保障するエージェント・サーバの構成図である。

【図２６】

メッセンジャ・オブジェクトの各処理状態を説明した表である。

【符号の説明】

１０　メッセージ処理システム

１２　ネットワーク

１４　メッセージ処理サーバ

１５　情報源クライアント

１６　処理要求元コンピュータ

２０　メッセージ処理装置

２１　処理要求元検索情報

２２　処理要求元検索情報管理手段

２３　エージェント

２４　永続記憶装置

２５　キャッシュ・メモリ

２７　受付手段

２８　リスト情報作成手段

３１　エージェント用指示手段

３２　繰返し指示手段

３４　メッセージ・ハンドラ

３５　データ

３８　挿入・読込み手段

３９　メッセージ・キュー

４２　メッセージ処理装置

４３　選択手段

４４　第１のメッセージ・キュー用処理機構

４５　第２のメッセージ・キュー用処理機構

５２　メッセージ処理装置

５３　処理要求元決定手段

５４　サブ処理優先度決定手段

５５　複合処理優先度決定手段

５６　エージェント用指示手段

５９　エージェント管理手段

６０　存在検出手段

６１　グループ化情報管理手段

６２　更新指示手段

６５　受付順番情報管理手段

６６　割当て手段

６７　スレッド

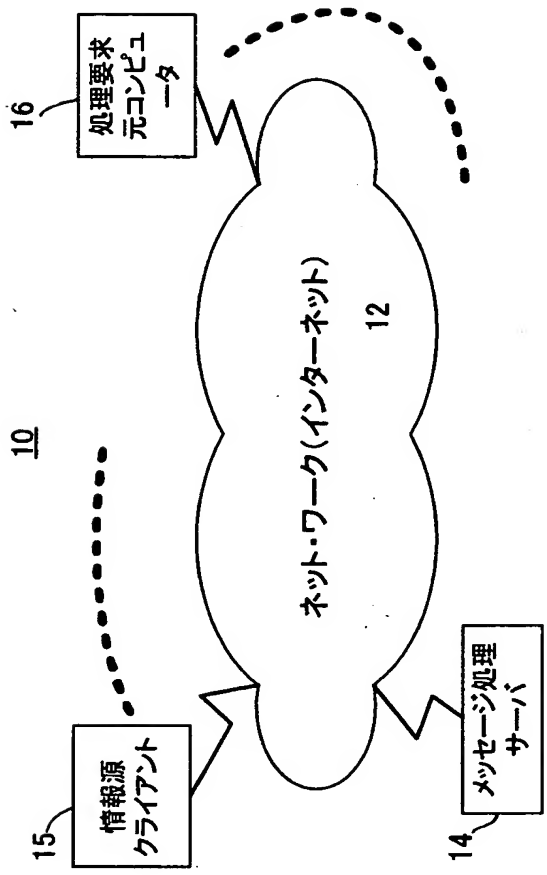
７０　進行状態情報管理手段

７２　判定手段

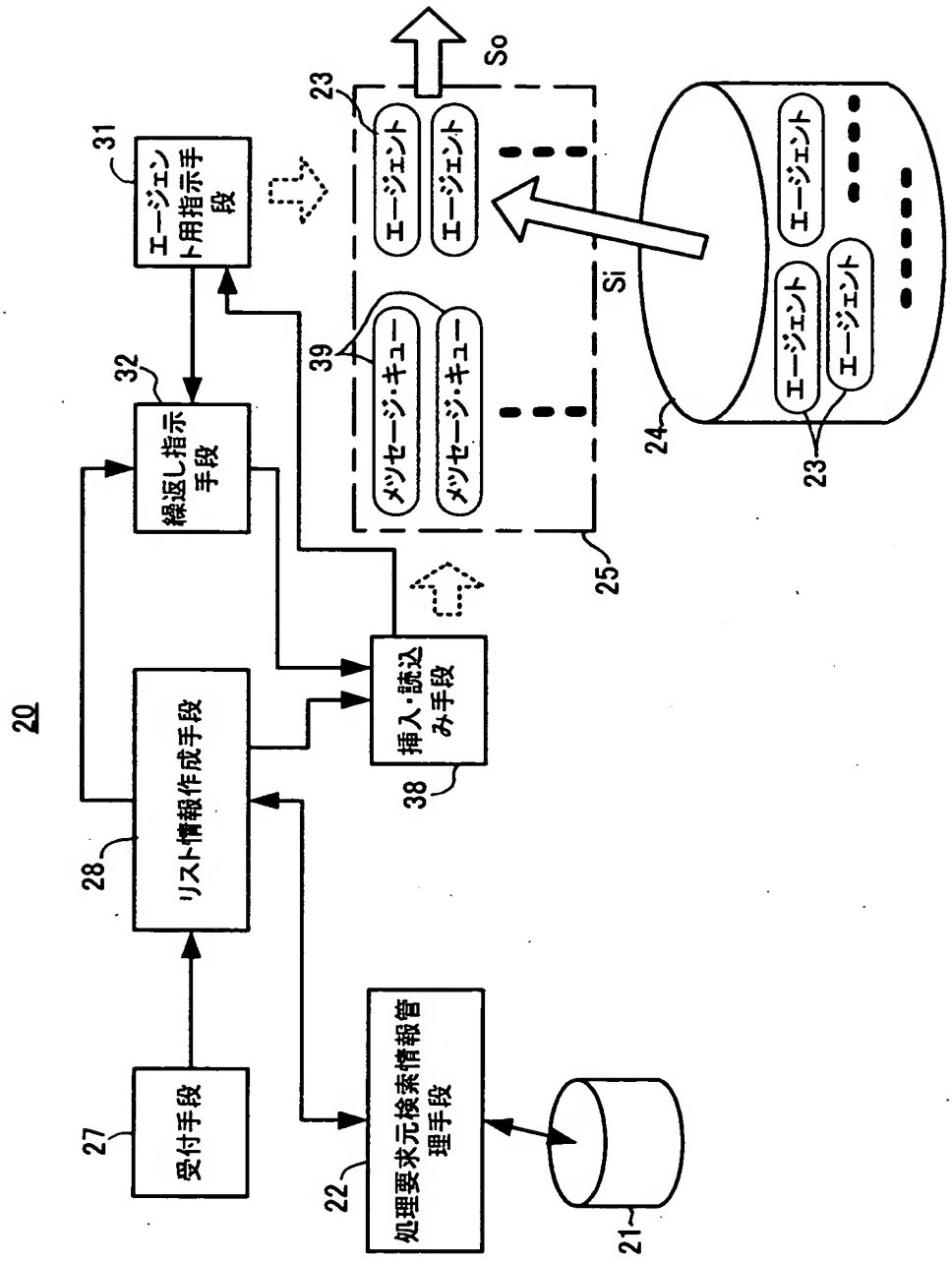
７３　エージェント制御手段

【書類名】　　図面

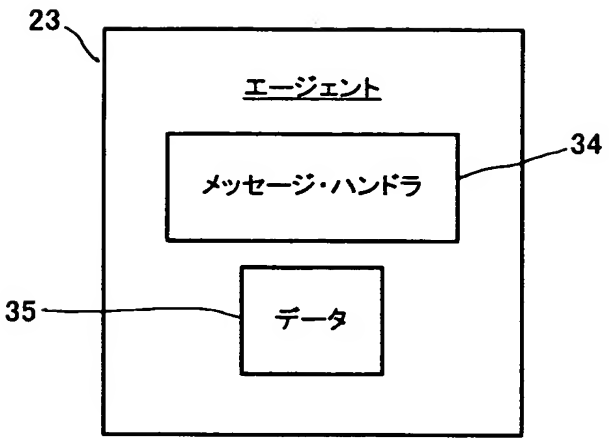
【図１】


処理要求元コンピュータ
16
ネット・ワーク（インターネット）
12
10
情報源クライアント
15
メッセージ処理サーバ
14

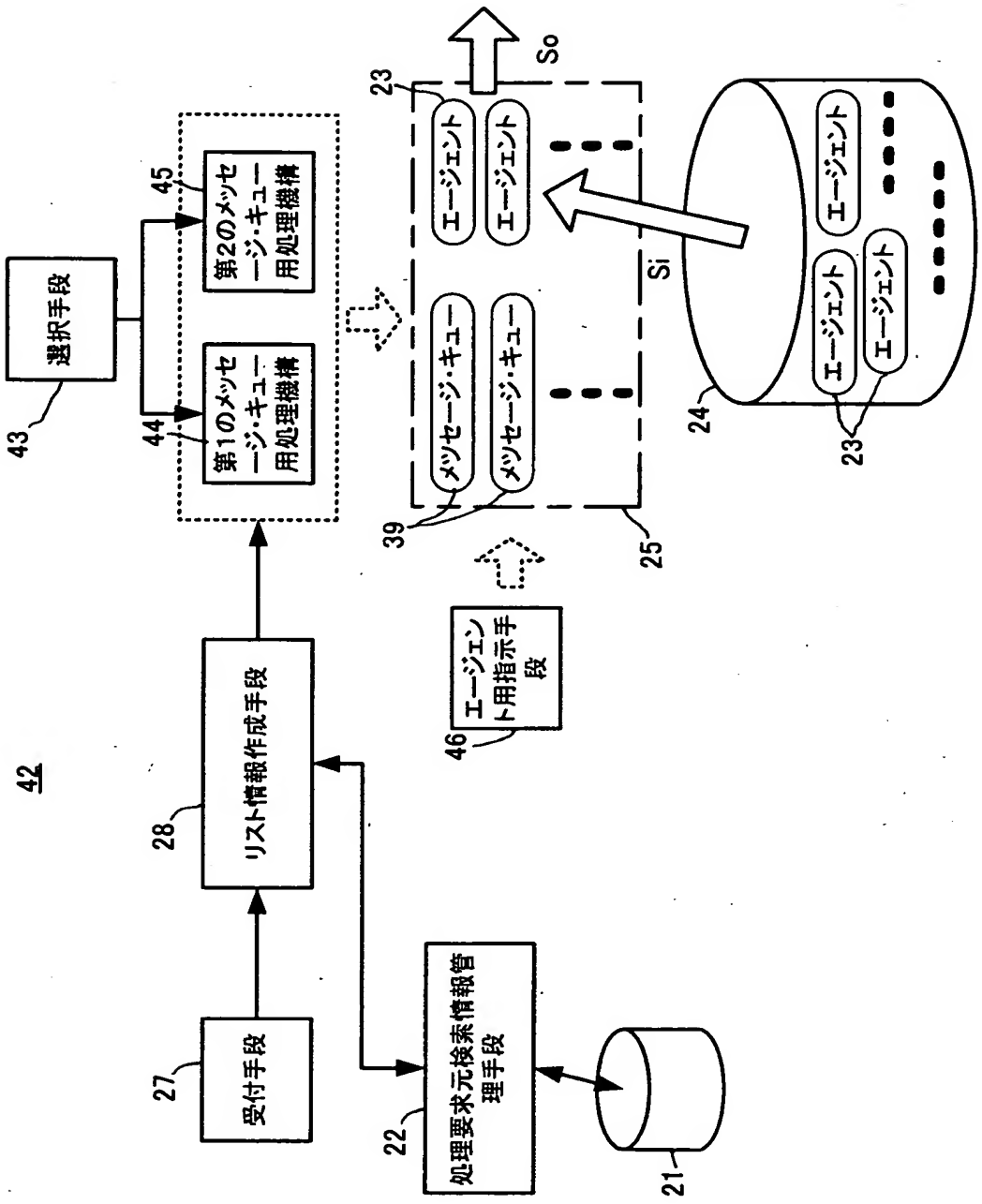
【図２】


20
エージェント用指示手段
31
繰返し指示手段
32
リスト情報作成手段
28
挿入・読込み手段
38
受付手段
27
処理要求元検索情報管理手段
22
21
23
エージェント
エージェント
39
メッセージ・キュー
メッセージ・キュー
25
So
Si
24
23
エージェント
エージェント
エージェント

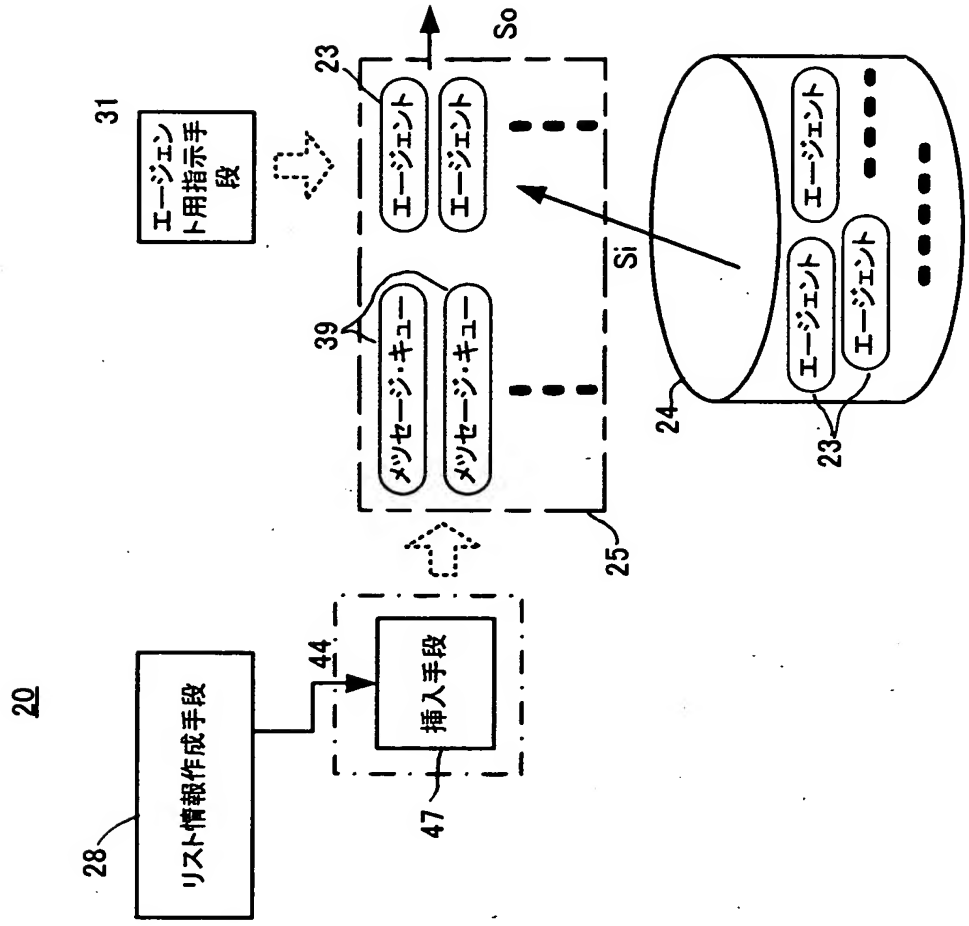
【図３】



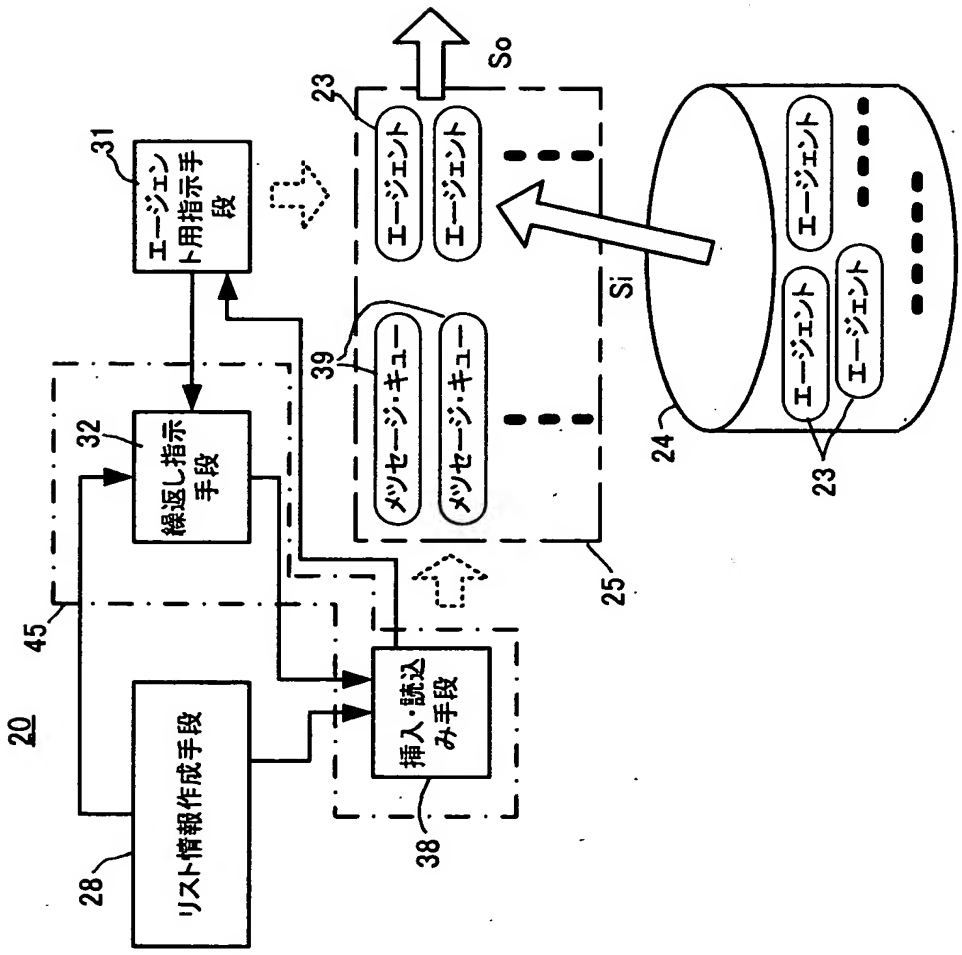
【図４】


42
選択手段
43
第1のメッセージ・キュー用処理機構
44
第2のメッセージ・キュー用処理機構
45
リスト情報作成手段
28
受付手段
27
処理要求元検索情報管理手段
22
21
エージェント用指示手段
46
メッセージ・キュー
メッセージ・キュー
39
エージェント
エージェント
23
So
Si
25
エージェント
エージェント
エージェント
23
24

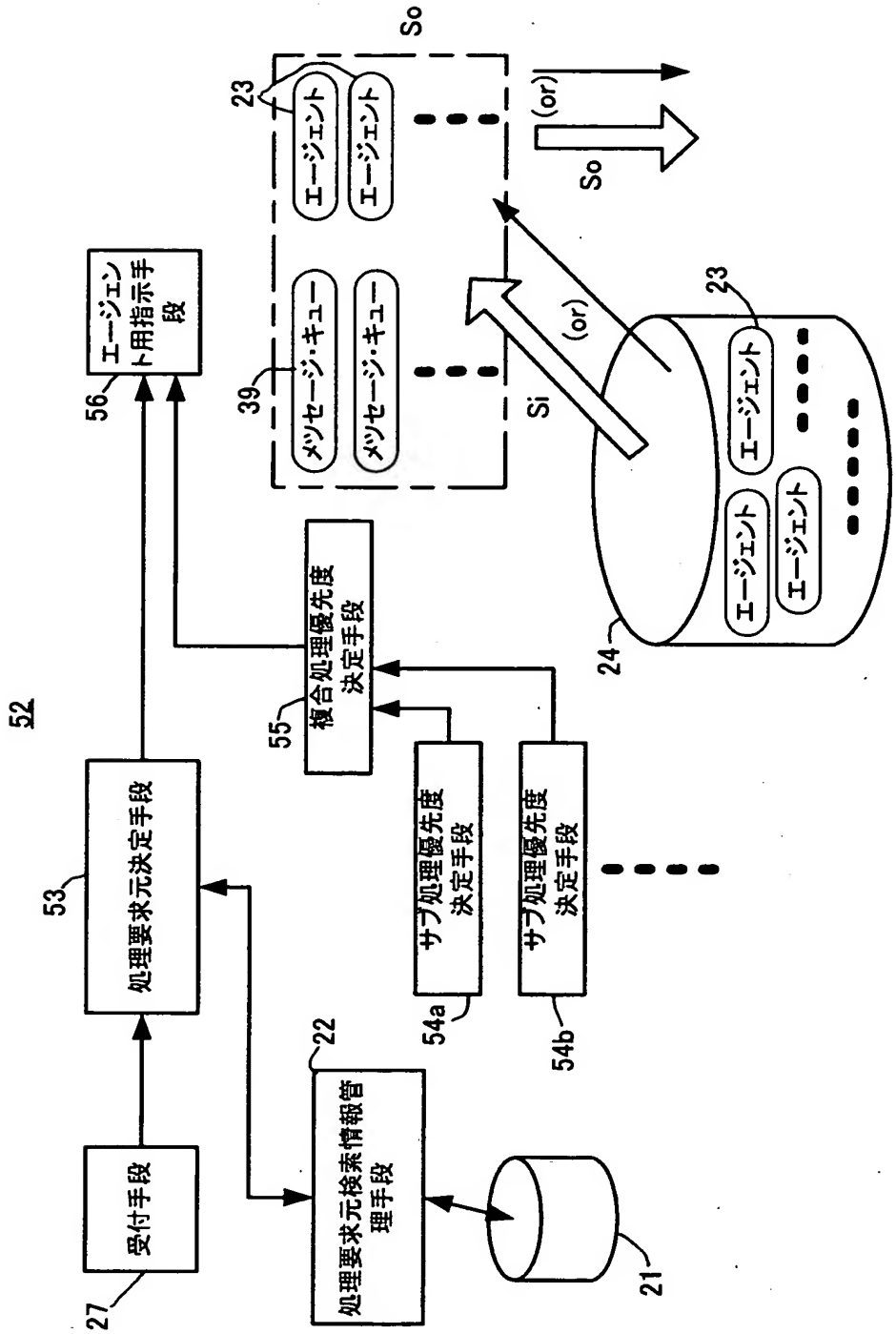
【図５】


20
リスト情報作成手段
28
挿入手段
44
47
エージェント用指示手段
31
23
エージェント
エージェント
So
Si
39
メッセージ・キュー
メッセージ・キュー
25
24
23
エージェント
エージェント
エージェント

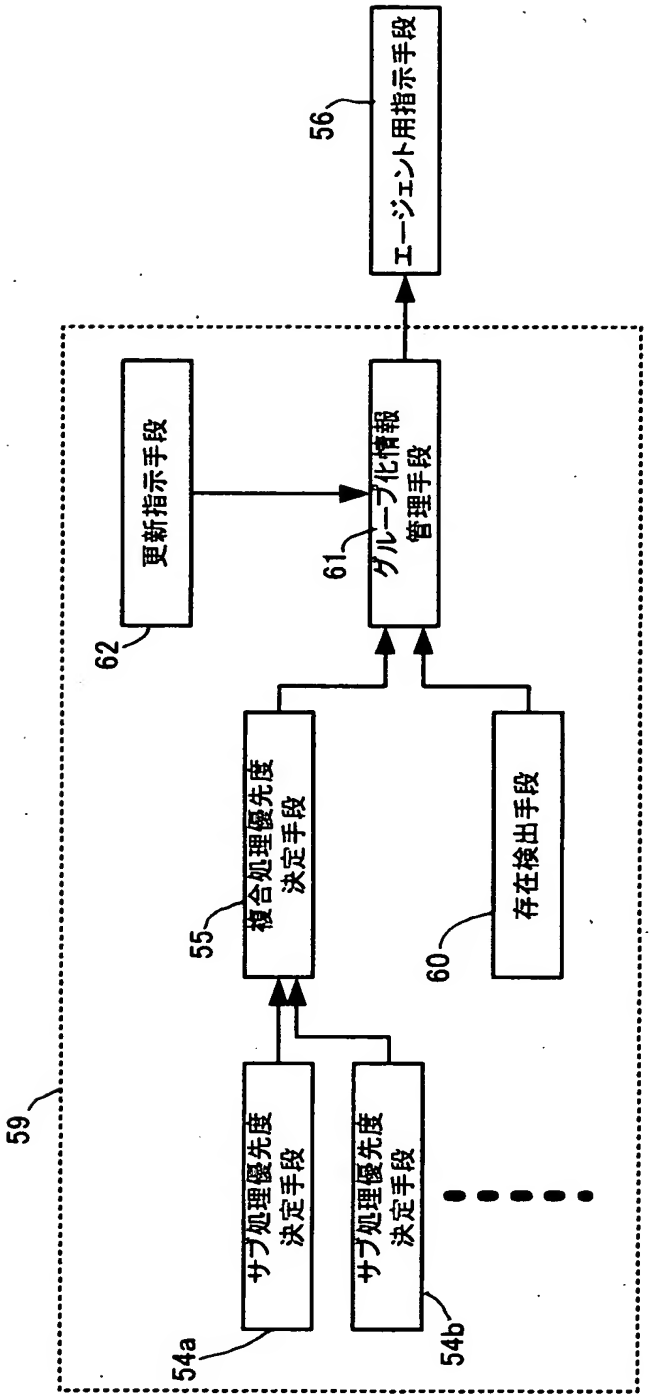
【図６】


So
23
エージェント
エージェント
31
エージェント用指示手段
Si
39
メッセージ・キュー
メッセージ・キュー
32
繰返し指示手段
24
23
エージェント
エージェント
エージェント
エージェント
25
45
20
挿入・読込み手段
38
リスト情報作成手段
28

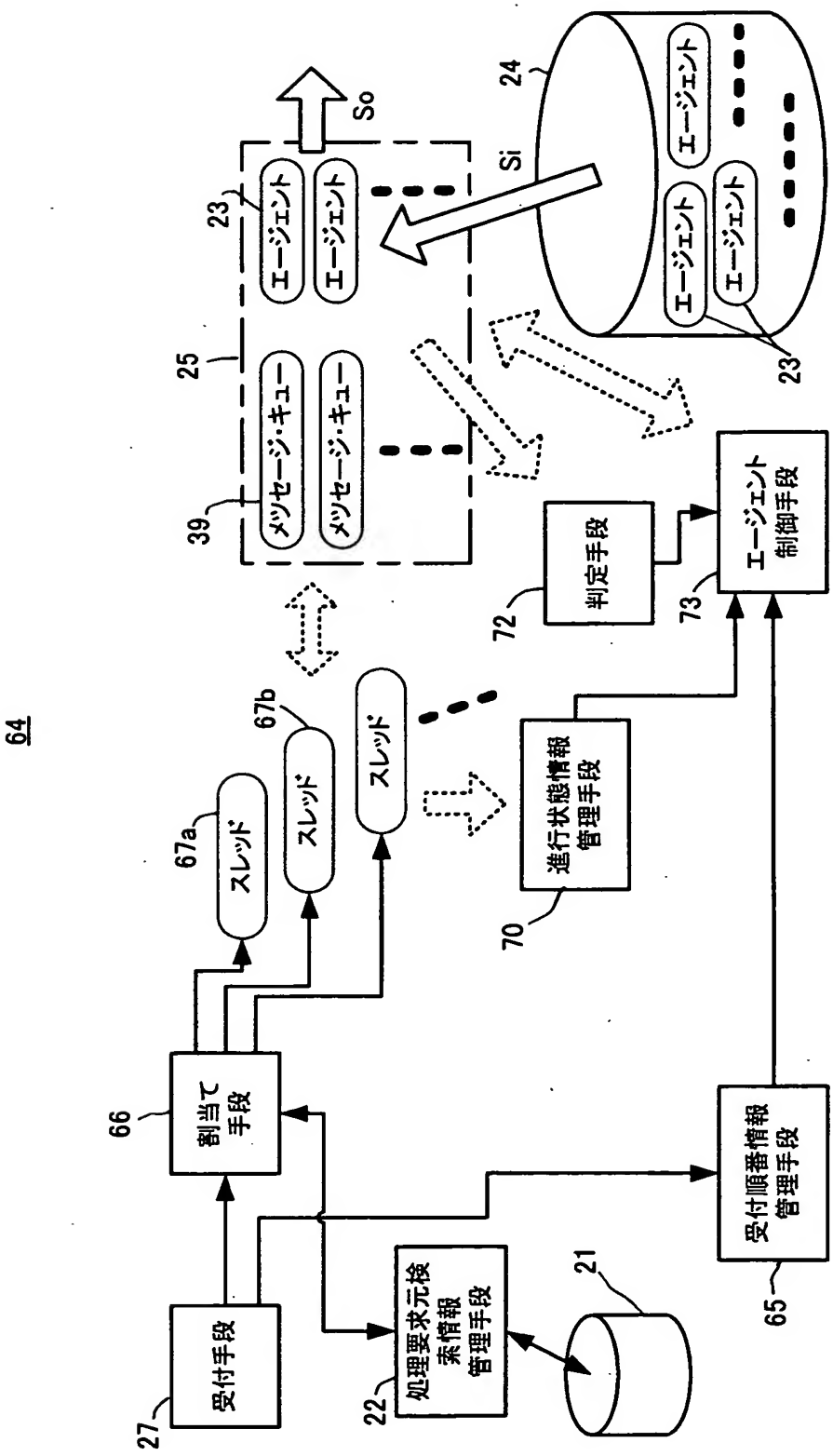
【図７】


52
受付手段
27
処理要求元検索情報管理手段
22
21
処理要求元決定手段
53
エージェント用指示手段
56
サブ処理優先度決定手段
54a
サブ処理優先度決定手段
54b
複合処理優先度決定手段
55
メッセージ・キュー
メッセージ・キュー
39
エージェント
エージェント
23
So
(or)
So
Si
(or)
23
エージェント
エージェント
エージェント
24

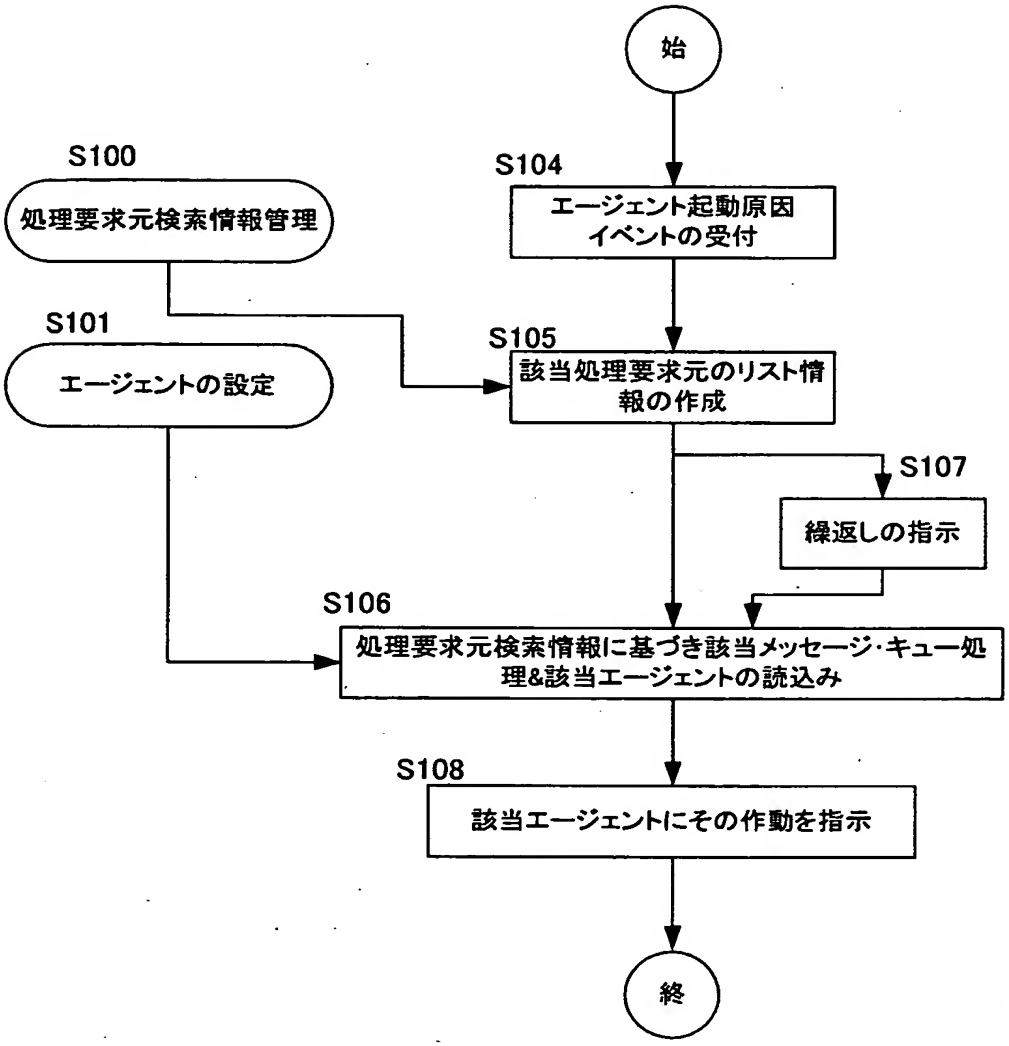
【図８】


56
エージェント用指示手段
59
更新指示手段
62
61
グループ化情報
管理手段
55
複合処理優先度
決定手段
60
存在検出手段
サブ処理優先度
決定手段
54a
サブ処理優先度
決定手段
54b

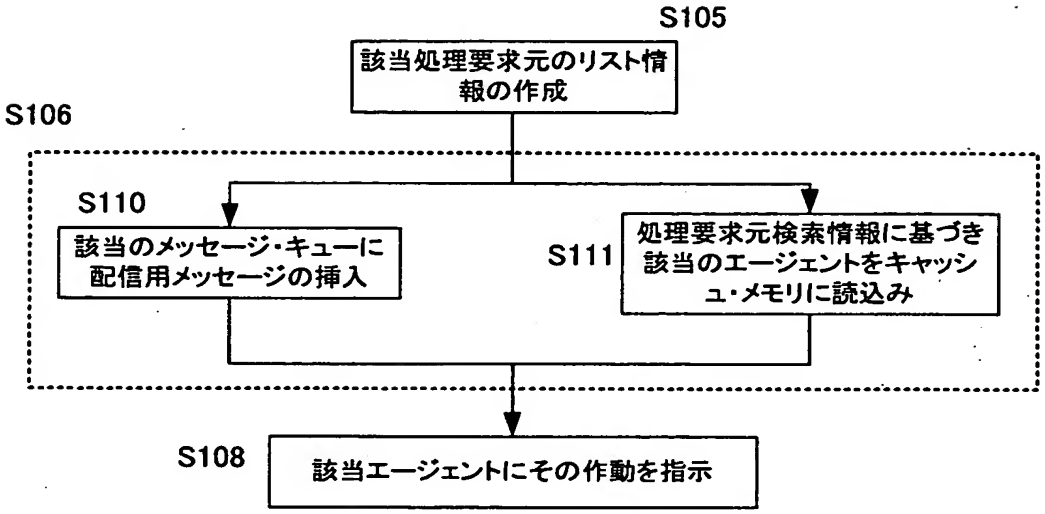
【図９】


64
受付手段
27
処理要求元検索情報管理手段
22
21
割当て手段
66
受付順番情報管理手段
65
スレッド
67a
スレッド
67b
スレッド
進行状態情報管理手段
70
判定手段
72
エージェント制御手段
73
メッセージ・キュー
メッセージ・キュー
39
25
エージェント
エージェント
23
So
Si
24
エージェント
エージェント
エージェント
23

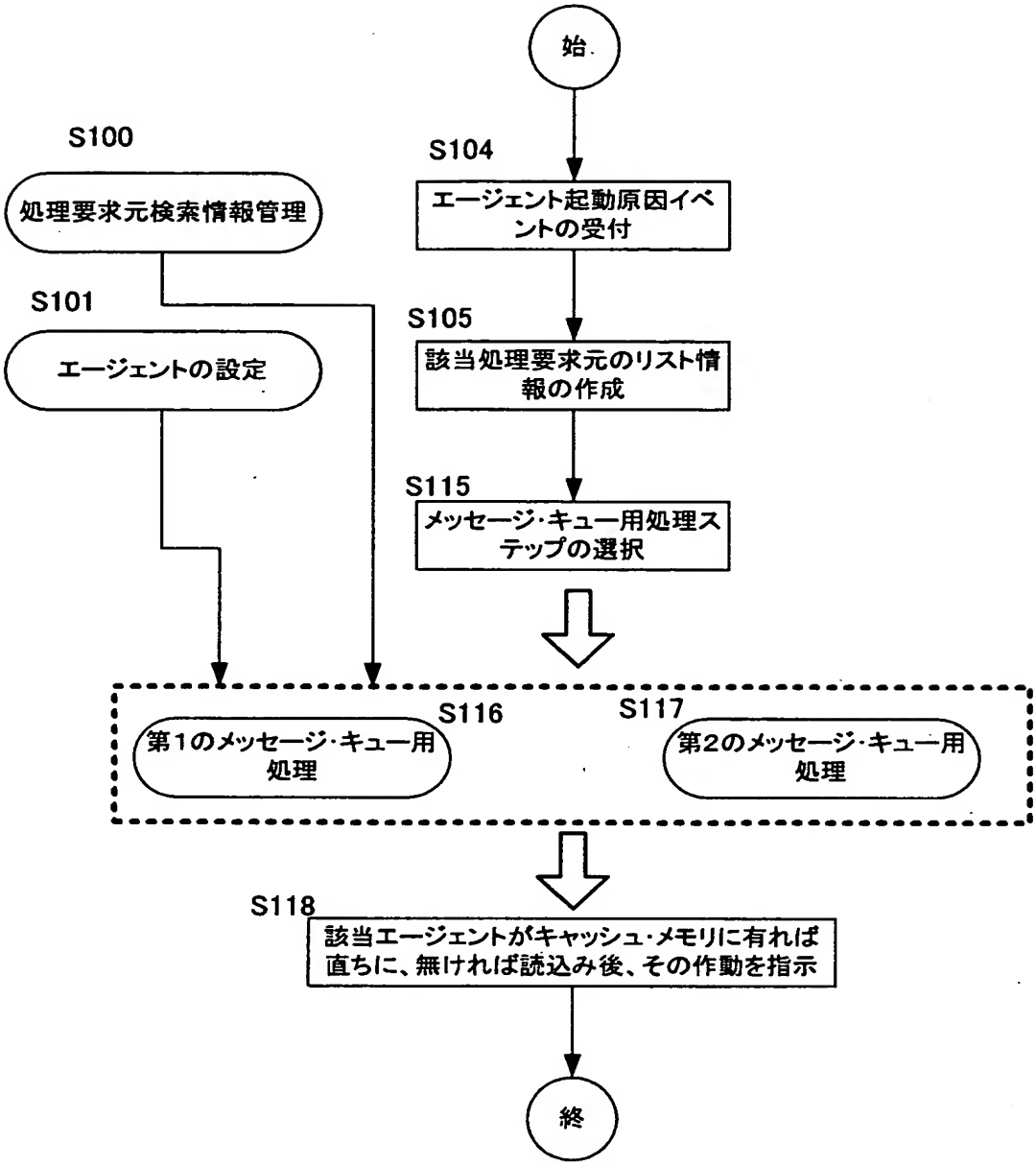
【図１０】


始
S100
処理要求元検索情報管理
S104
エージェント起動原因
イベントの受付
S101
エージェントの設定
S105
該当処理要求元のリスト情
報の作成
S107
繰返しの指示
S106
処理要求元検索情報に基づき該当メッセージ･キュー処
理&該当エージェントの読込み
S108
該当エージェントにその作動を指示
終

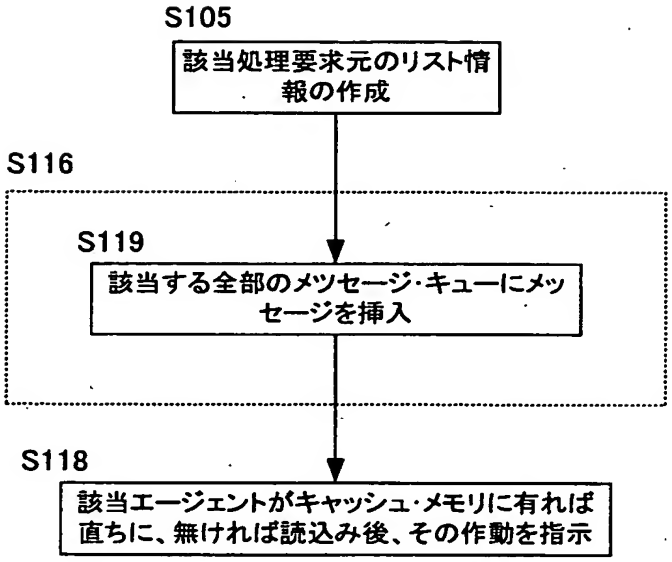
【図１１】


S105
該当処理要求元のリスト情報の作成
S106
S110
該当のメッセージ・キューに配信用メッセージの挿入
S111
処理要求元検索情報に基づき該当のエージェントをキャッシュ・メモリに読込み
S108
該当エージェントにその作動を指示

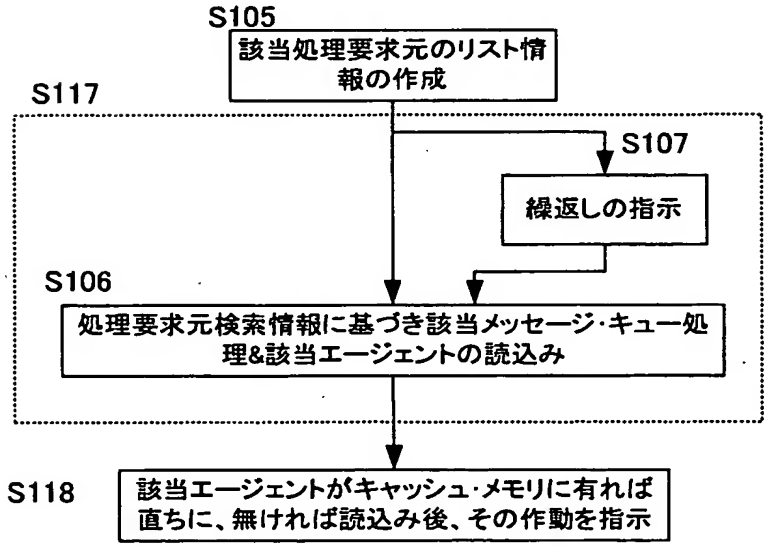
【図１２】


始
S100
処理要求元検索情報管理
S104
エージェント起動原因イベントの受付
S101
エージェントの設定
S105
該当処理要求元のリスト情報の作成
S115
メッセージ･キュー用処理ステップの選択
S116
第1のメッセージ･キュー用処理
S117
第2のメッセージ･キュー用処理
S118
該当エージェントがキャッシュ･メモリに有れば直ちに、無ければ読込み後、その作動を指示
終

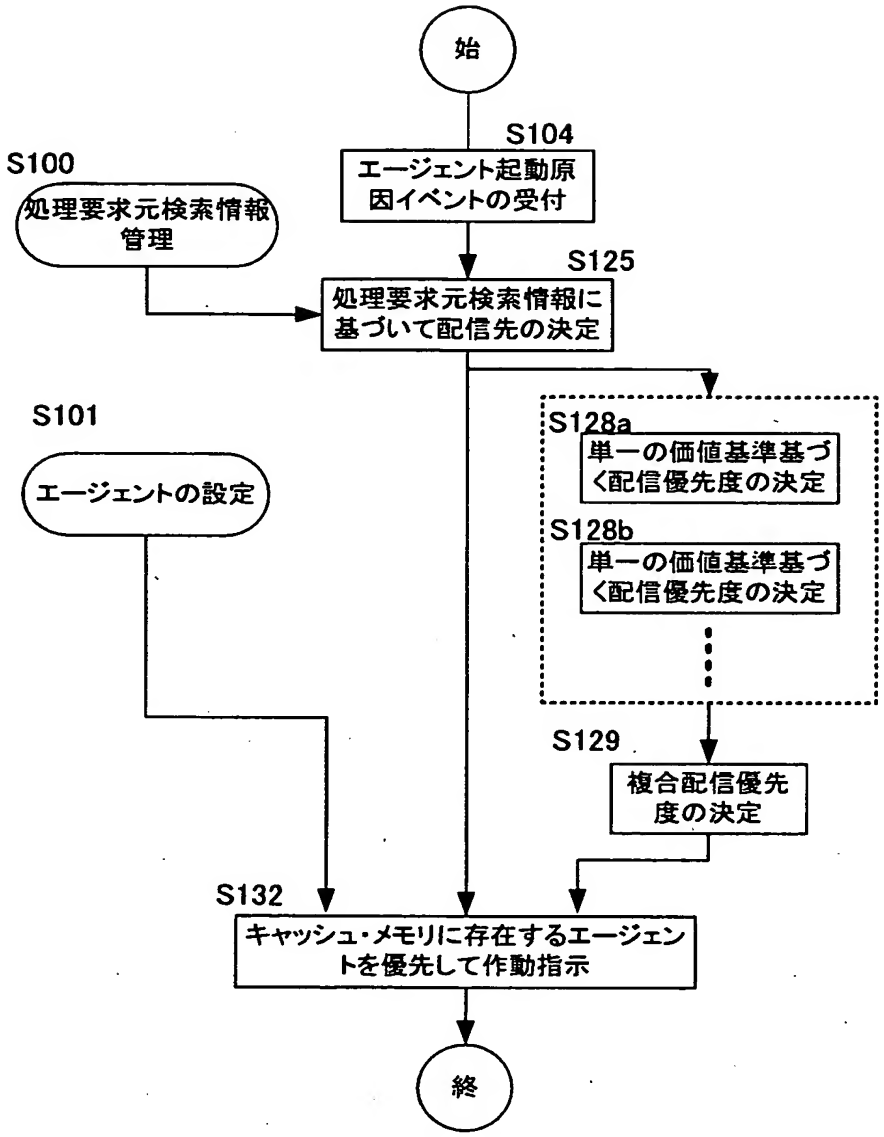
【図１３】


S105
該当処理要求元のリスト情報の作成
S116
S119
該当する全部のメッセージ・キューにメッセージを挿入
S118
該当エージェントがキャッシュ・メモリに有れば直ちに、無ければ読込み後、その作動を指示

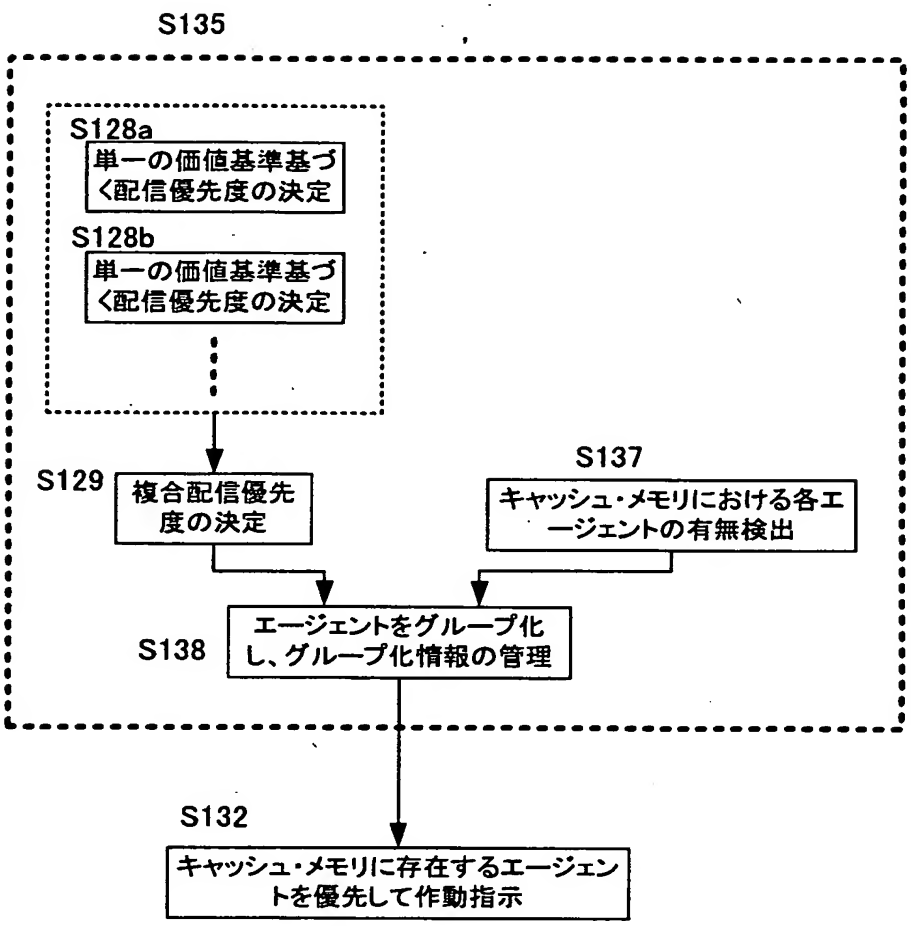
【図１４】


S105
該当処理要求元のリスト情報の作成
S117
S107
繰返しの指示
S106
処理要求元検索情報に基づき該当メッセージ・キュー処理&該当エージェントの読込み
S118
該当エージェントがキャッシュ・メモリに有れば直ちに、無ければ読込み後、その作動を指示

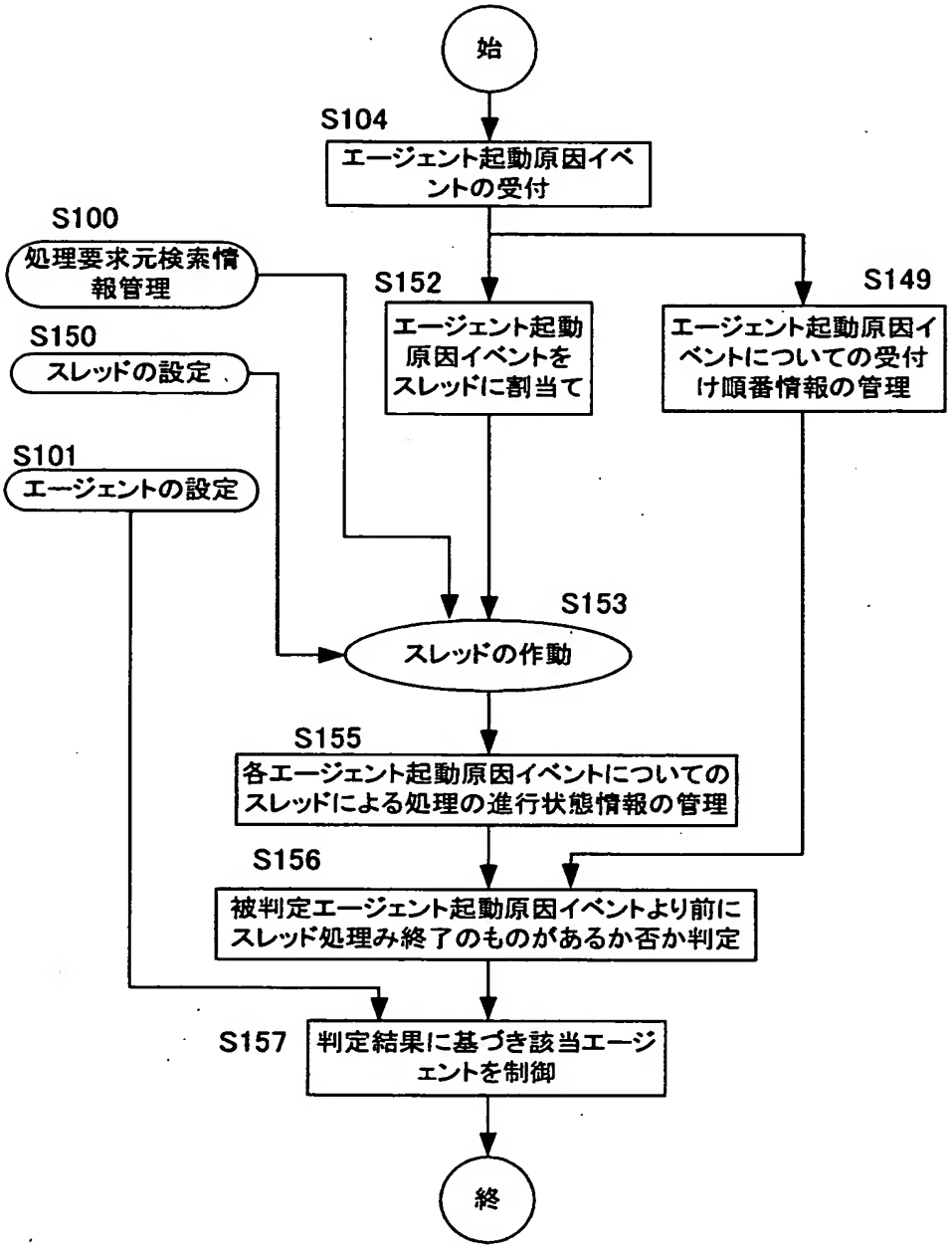
【図１５】


始
S104
エージェント起動原因イベントの受付
S100
処理要求元検索情報管理
S125
処理要求元検索情報に基づいて配信先の決定
S101
エージェントの設定
S128a
単一の価値基準基づく配信優先度の決定
S128b
単一の価値基準基づく配信優先度の決定
S129
複合配信優先度の決定
S132
キャッシュ・メモリに存在するエージェントを優先して作動指示
終

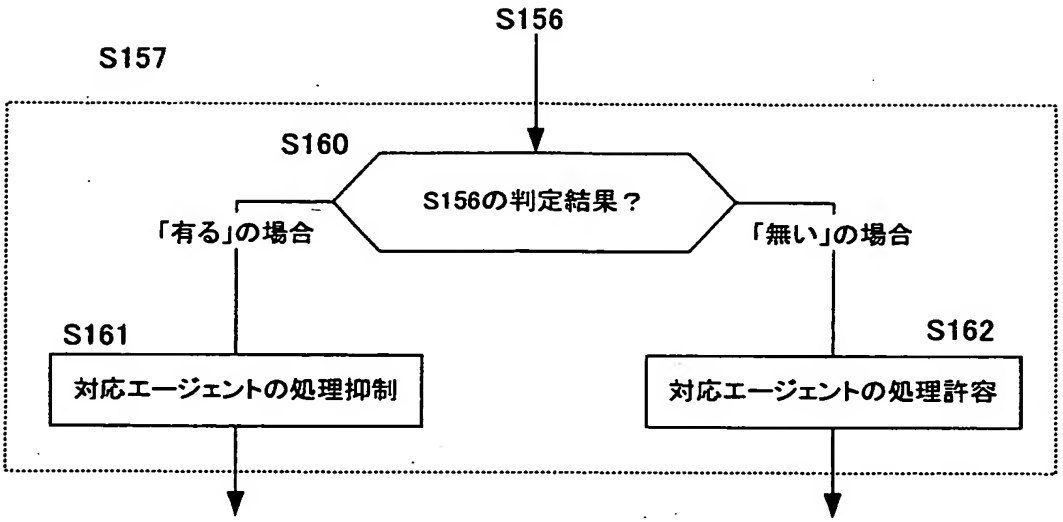
【図１６】


S135
S128a
単一の価値基準基づく配信優先度の決定
S128b
単一の価値基準基づく配信優先度の決定
S129
複合配信優先度の決定
S137
キャッシュ・メモリにおける各エージェントの有無検出
S138
エージェントをグループ化し、グループ化情報の管理
S132
キャッシュ・メモリに存在するエージェントを優先して作動指示

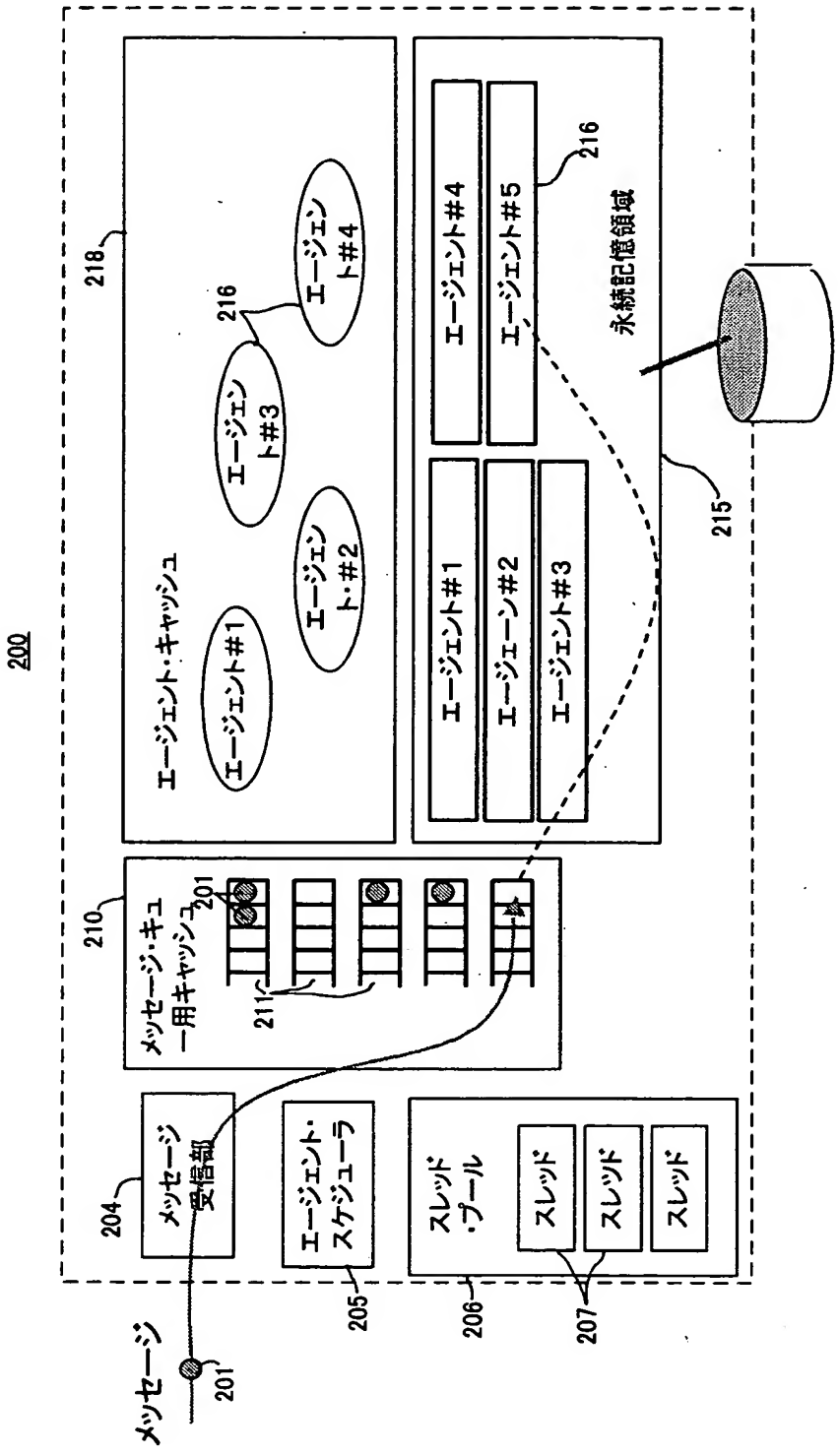
【図１７】


始
S104
エージェント起動原因イベントの受付
S100
処理要求元検索情報管理
S152
エージェント起動原因イベントをスレッドに割当て
S149
エージェント起動原因イベントについての受付け順番情報の管理
S150
スレッドの設定
S101
エージェントの設定
S153
スレッドの作動
S155
各エージェント起動原因イベントについてのスレッドによる処理の進行状態情報の管理
S156
被判定エージェント起動原因イベントより前にスレッド処理み終了のものがあるか否か判定
S157
判定結果に基づき該当エージェントを制御
終

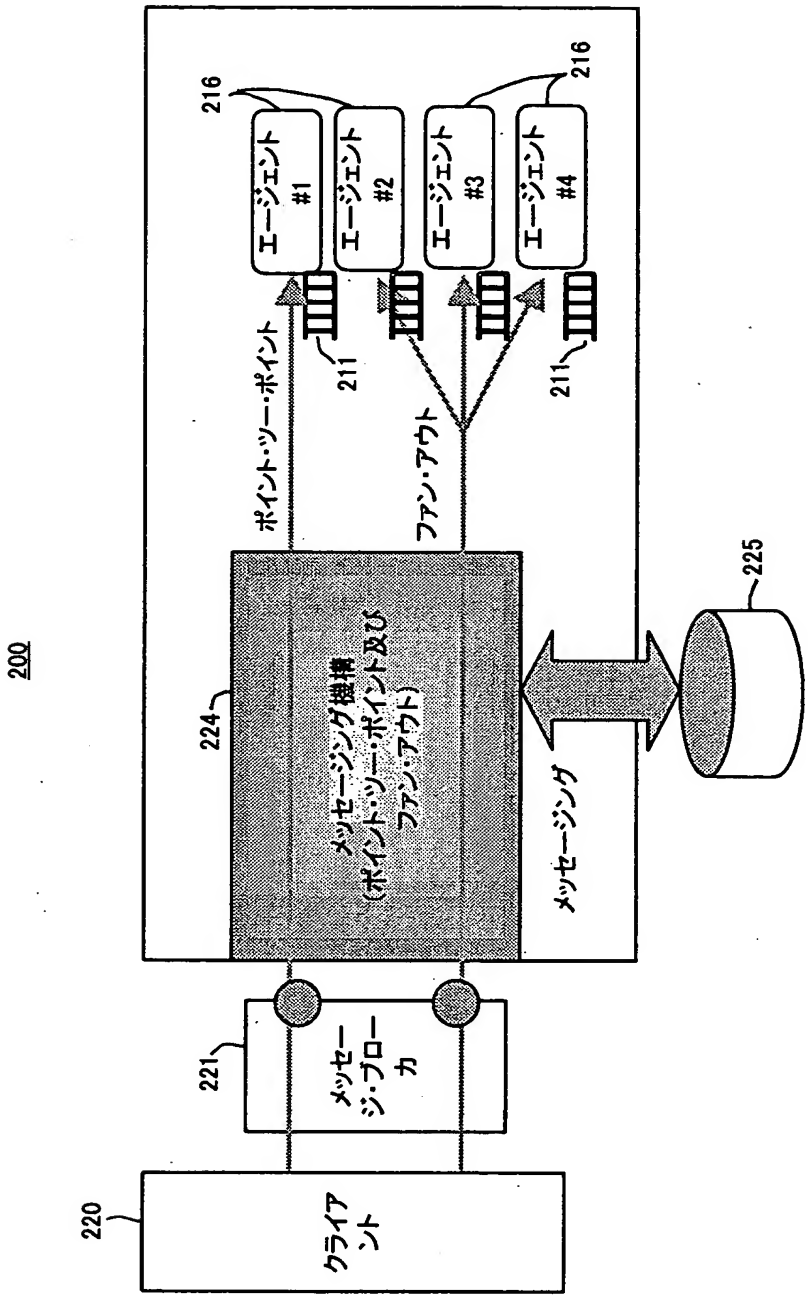
【図１８】


S156
S157
S160
S156の判定結果？
「有る」の場合
「無い」の場合
S161
対応エージェントの処理抑制
S162
対応エージェントの処理許容

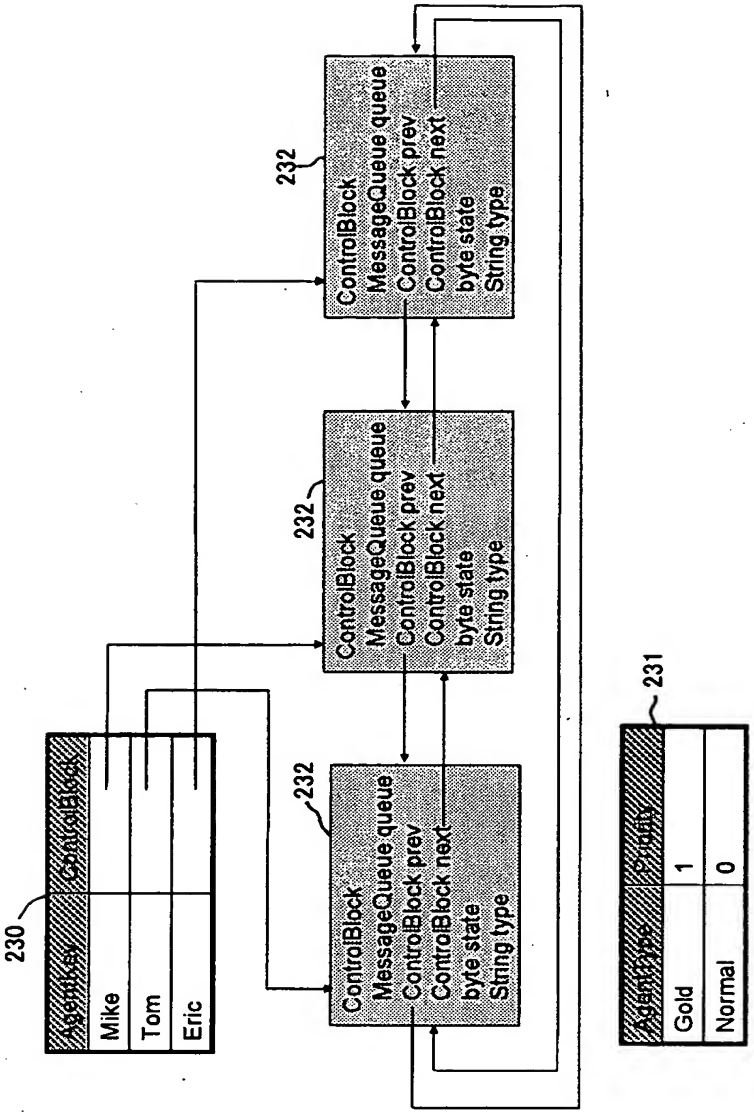
【図１９】


200
218
エージェント・キャッシュ
エージェント#1
エージェント#2
エージェント#3
エージェント#4
216
永続記憶領域
エージェント#1
エージェント#2
エージェント#3
エージェント#4
エージェント#5
216
215
210
メッセージ・キュー用キャッシュ
201
211
204
メッセージ受信部
エージェント・スケジューラ
スレッド・プール
スレッド
スレッド
スレッド
205
206
207
メッセージ
201

【図２０】


200
220
クライアント
221
メッセージ・ブローカ
224
メッセージング機構
(ポイント・ツー・ポイント及び
ファン・アウト)
225
メッセージング
ポイント・ツー・ポイント
ファン・アウト
211
211
216
216
エージェント #1
エージェント #2
エージェント #3
エージェント #4

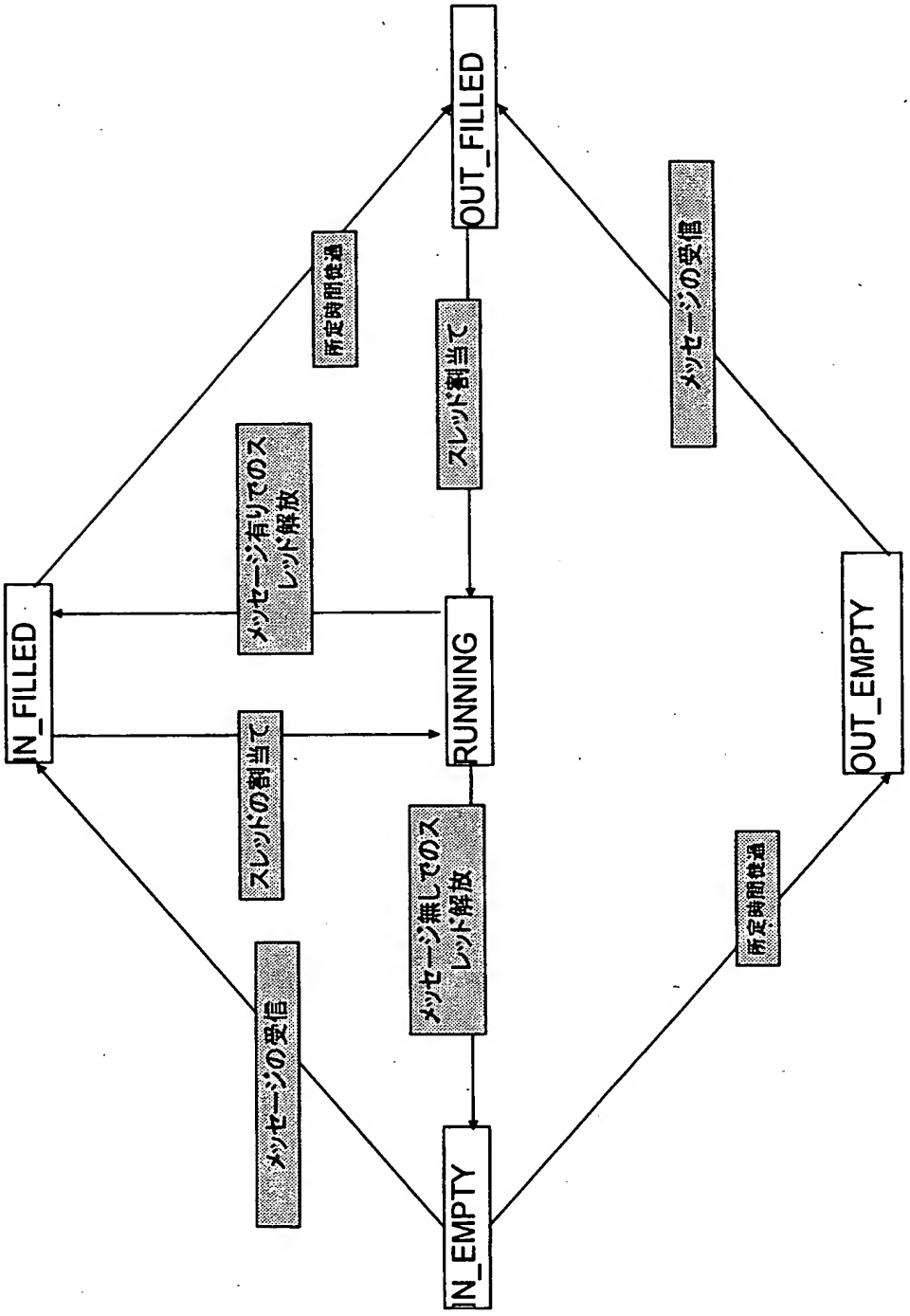
【図２１】


230
AgentKey
ControlBlock
Mike
Tom
Eric
232
ControlBlock
MessageQueue queue
ControlBlock prev
ControlBlock next
byte state
String type
231
AgentType
Point
Gold
1
Normal
0

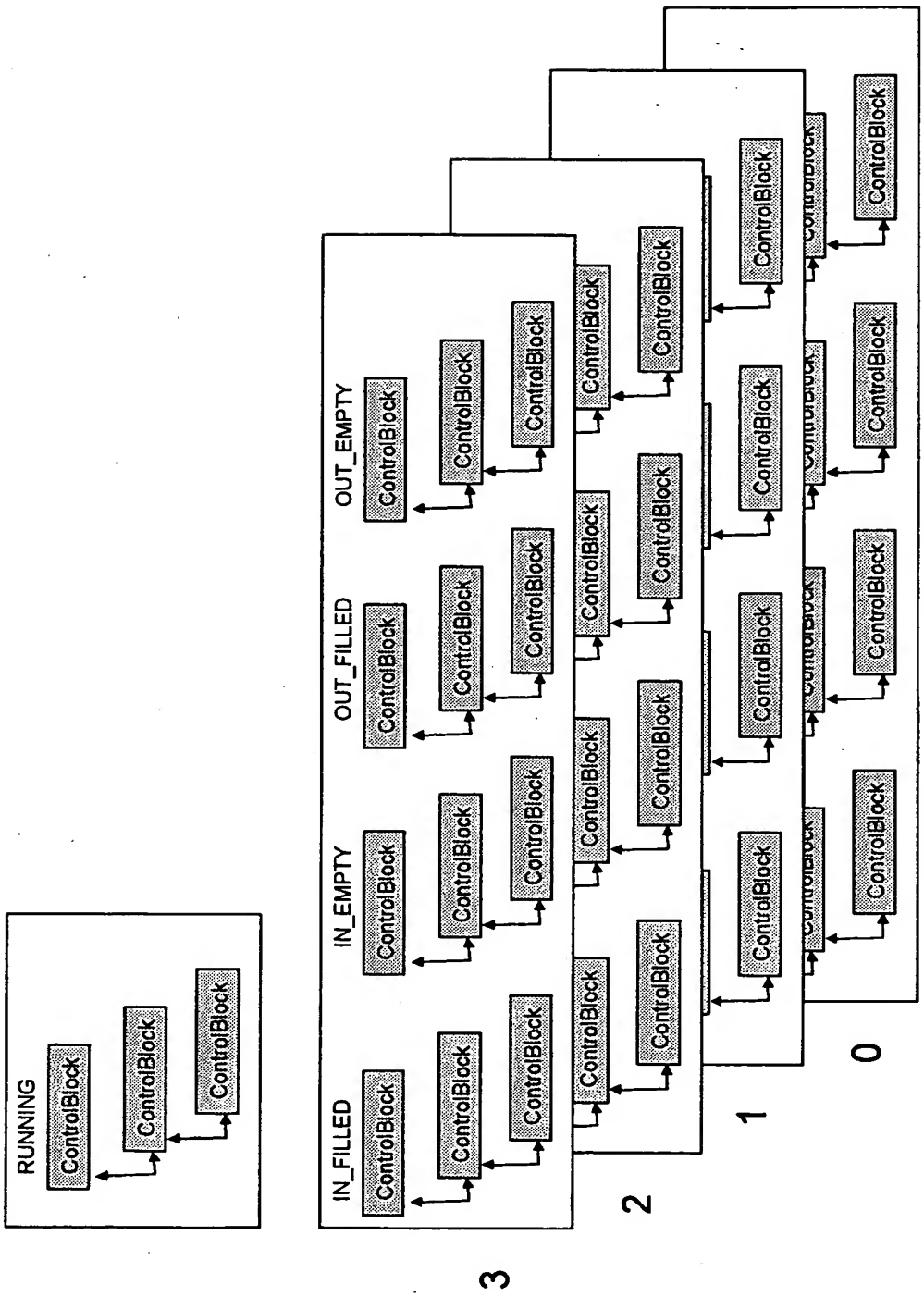
【図２２】

RUNNING	スレッドが割り当てられている
IN_FILLED	エージェント・キャッシュにあり、メッセージ・キューにメッセージがある
IN_EMPTY	エージェント・キャッシュにあり、メッセージ・キューが空である
OUT_FILLED	エージェント・キャッシュになく、メッセージ・キューにメッセージがある。
OUT_EMPTY	エージェント・キャッシュになく、メッセージ・キューが空である。

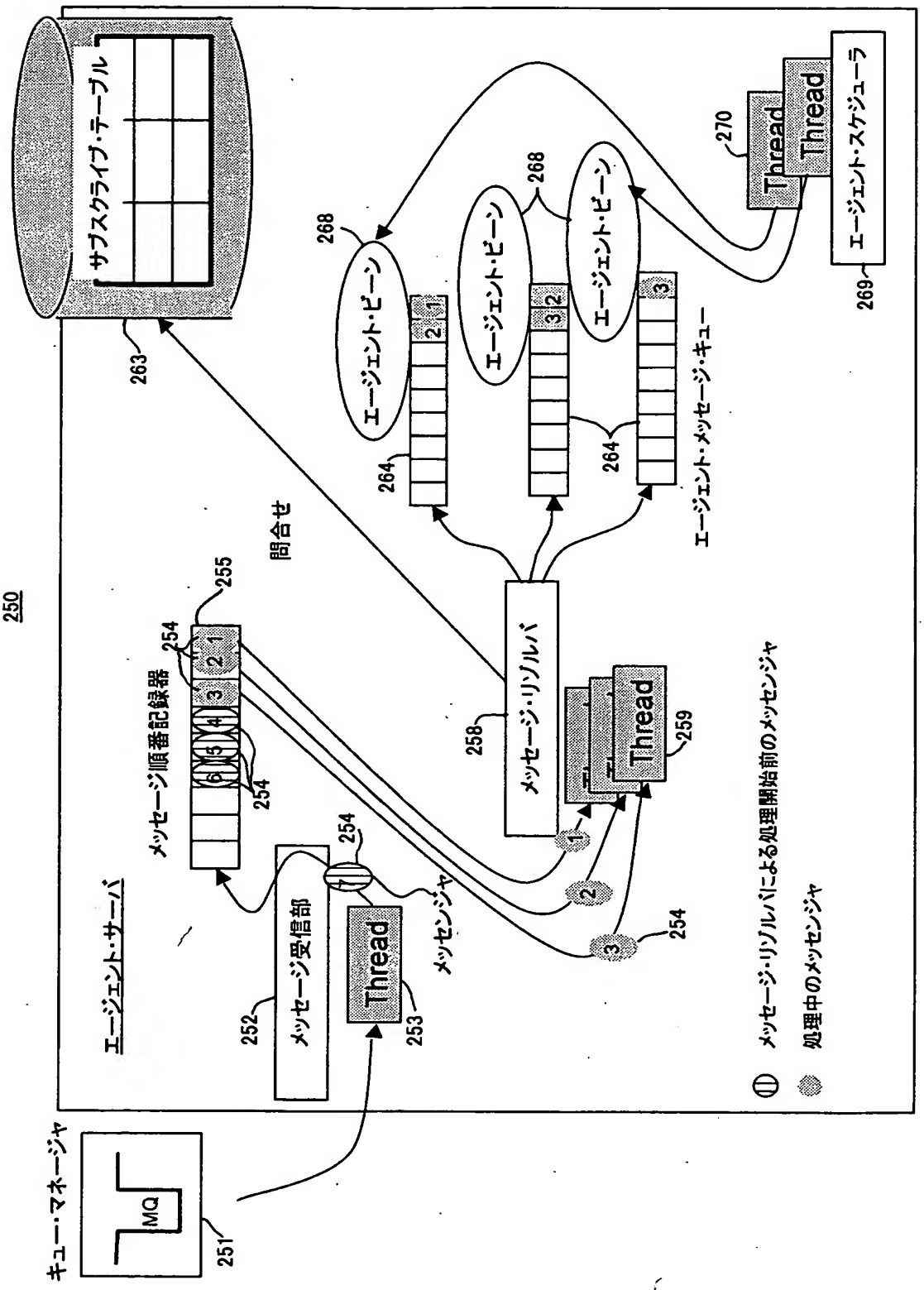
【図２３】


OUT_FILLED
所定時間経過
スレッド割当て
メッセージの受信
メッセージ有りでのスレッド解放
IN_FILLED
RUNNING
OUT_EMPTY
スレッドの割当て
メッセージ無しでのスレッド解放
所定時間経過
メッセージの受信
IN_EMPTY

【図２４】


RUNNING
ControlBlock
ControlBlock
ControlBlock
IN_FILLED
IN_EMPTY
OUT_FILLED
OUT_EMPTY
3
2
1
0

【図２５】


キュー・マネージャ
MQ
251
エージェント・サーバ
250
メッセージ受信部
252
Thread
253
メッセージジャ
254
メッセージ順番記録器
255
問合せ
263
サブスクライブ・テーブル
メッセージ・リソルバ
258
Thread
259
エージェント・メッセージ・キュー
264
エージェント・ビーン
268
エージェント・スケジューラ
269
Thread
270
メッセージ・リソルバによる処理開始前のメッセンジャ
処理中のメッセンジャ

【図２６】

NotProcessed (未処理)	MessageOrderRecorder に記録があり、MessageResolver により処理されていない状態を表す
Distributing (処理中)	MessageOrderRecorder に記録があり、MessageResolver により現在処理中である状態を表す
Distributed (処理終了)	MessageOrderRecorder に記録があり、MessageResolver により処理が完了され、すべてのあて先エージェントのメッセージキューに挿入されている状態を表す
Ready (準備完了)	エージェントが処理を行っても良い状態を表す。この状態になるのは、MessageOrderRecorder の記録からすでに削除されているか、または、MessageOrderRecorder の記録の先頭にあるかのどちらかである。

【書類名】　　要約書

【要約】

【課題】　数十万～数百万等の膨大な処理要求元に適用するメッセージを、メッセージに対応のエージェントを使って所定するときの処理時間を短縮する。

【解決手段】　リスト情報作成手段２８は、エージェント起動原因イベントに対して該当処理要求元についてのリスト情報を処理要求元検索情報２１に基づいて作成する。挿入・読込み手段３８は、リスト情報に含まれる処理要求元の中から未選択の複数個を挿入・読込み対象処理要求元として選択し、該挿入・読込み対象処理要求元に係るメッセージ・キュー３９に、メッセージを挿入するとともに、挿入・読込み対象処理要求元に係るエージェント２３をキャッシュ・メモリ２５へ読込む、エージェント用指示手段３１は、メッセージが挿入されているメッセージ・キュー３９に係るエージェント２３にその作動を指示する。繰返し指示手段３２は挿入・読込み手段３８にその処理の繰返しを指示する。

【選択図】　　図２

認定・付加情報

特許出願の番号	特願２００２－３５５６４１
受付番号	５０２０１８５３４４６
書類名	特許願
担当官	小野寺　光子　　１７２１
作成日	平成１４年１２月　９日

＜認定情報・付加情報＞

【提出日】　平成14年12月 6日

【特許出願人】

【識別番号】　390009531

【住所又は居所】　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　ニュー　オーチャード　ロード

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】　100086243

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内

【氏名又は名称】　坂口　博

【代理人】

【識別番号】　100091568

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内

【氏名又は名称】　市位　嘉宏

【代理人】

【識別番号】　100108501

【住所又は居所】　神奈川県大和市下鶴間１６２３番１４　日本アイ・ビー・エム株式会社　知的所有権

【氏名又は名称】　上野　剛史

【復代理人】　申請人

【識別番号】　100085408

【住所又は居所】　東京都中央区日本橋２丁目１番１号　櫻正宗ビル９階

【氏名又は名称】　山崎　隆

次頁無

出　願　人　履　歴　情　報

識別番号　〔３９０００９５３１〕

１．変更年月日　２００２年　６月　３日
［変更理由］　住所変更
住　所　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
氏　名　インターナショナル・ビジネス・マシーンズ・コーポレーション